



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

DISEÑO E IMPLEMENTACIÓN DE ROBOT MÓVIL, PARA LA INSPECCIÓN DE LÍNEAS ELÉCTRICAS

ING. CARLOS GABRIEL EIBAR BEJARANO

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

MAGISTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN INDUSTRIAL

RIOBAMBA - ECUADOR

JULIO 2017

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, denominado: "DISEÑO E IMPLEMENTACIÓN DE ROBOT MÓVIL, PARA LA INSPECCIÓN DE LÍNEAS ELÉCTRICAS", de responsabilidad del señor Eibar Bejarano Carlos Gabriel, ha sido minuciosamente revisado y se autoriza su presentación.

Tribunal:

Ing. Wilson Armando Zúñiga Vinueza, MsC.

PRESIDENTE

Ing. Javier José Gavilanes Carrión, MsC.

TUTOR PROYECTO DE TITULACIÓN

Ing. Miguel Ángel Pérez Bayas, MsC.

MIEMBRO DEL TRIBUNAL

Ing. Édisson Iván Aldás Serrano, MsC.

MIEMBRO DEL TRIBUNAL

Riobamba, julio 2017

DERECHOS INTELECTUALES

Yo, Eibar Bejarano Carlos Gabriel soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica de Chimborazo.

EIBAR BEJARANO CARLOS GABRIEL

Nº Cédula: 180423747-5

DERECHOS DE AUTOR

Yo, Eibar Bejarano Carlos Gabriel, declaro que el presente proyecto de investigación, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

EIBAR BEJARANO CARLOS GABRIEL

Nº Cédula: 180423747-5

DEDICATORIA

El presente trabajo se lo dedico a mi esposa y mi hija, ya que son ellas mi puntal de apoyo y quienes me han brindado su fuerza y cariño para poder seguir sin decaer y poder realizar el presente proyecto para llegar a su exitosa culminación.

AGRADECIMIENTO

Agradezco infinitamente a Dios por darme la oportunidad de poder cruzar una etapa más de mi vida profesional, además a mis padres que siempre me han apoyado en cada uno de los pasos en el transcurso de mi vida, así como también mis hermanas y en general mi familia que siempre ha estado apoyándome de una u otra manera.

CONTENIDO

<i>CERTIFICACIÓN:</i>	<i>ii</i>
<i>DERECHOS DE AUTOR</i>	<i>iv</i>
<i>DEDICATORIA</i>	<i>v</i>
<i>AGRADECIMIENTO</i>	<i>vi</i>
<i>CONTENIDO</i>	<i>vii</i>
<i>ÍNDICE DE TABLAS</i>	<i>ix</i>
<i>ÍNDICE DE FIGURAS</i>	<i>x</i>
<i>RESUMEN</i>	<i>xiii</i>
<i>SUMMARY</i>	<i>xiv</i>

CAPÍTULO I

1.	INTRODUCCIÓN	1
1.1.	Título	1
1.2.	Planteamiento de problema	1
1.3.	Formulación del problema	1
1.4.	Sistematización del problema	1
1.5.	Objetivos de la investigación.....	2
1.5.1.	Objetivo general.....	2
1.5.2.	Objetivos específicos.....	2
1.6.	Justificación de la investigación	3
1.7.	Hipótesis.....	4

CAPÍTULO II

2.	MARCO DE REFERENCIA.....	5
2.1.	Estructuras de las líneas eléctricas.....	5
2.2.	Inspección y mantenimiento de líneas eléctricas	6
2.3.	Características de las líneas de distribución de energía en el Ecuador.....	8
2.4.	Procedimiento manual de inspección y mantenimiento de líneas eléctricas.....	9
2.5.	Robots existentes para inspección de líneas eléctricas	10
2.5.1.	Proyecto pelicano.....	10
2.5.2.	Linescout	11
2.5.3.	Robot PLIR	12
2.5.4.	Robtet.....	13
2.5.5.	Arman.....	13
2.5.6.	Snake-like	14
2.6.	Tipos de comunicación de robot de inspección de líneas eléctricas	15
2.7.	Módulos de comunicación xbee	16
2.8.	Módulo gps.....	17
2.9.	Arduino mega 2560.....	19
2.10.	Sensor infrarrojo de temperatura	20
2.10.1.	Funcionamiento sensor MLX90614.....	20
2.11.	Comunicación i2c	22

2.12.	Sensor ultrasónico HC-SR04	24
-------	----------------------------------	----

CAPÍTULO III

3.	DISEÑO E IMPLEMENTACIÓN	26
3.1.	Diseño estructura	27
3.1.1.	Cálculo torque de brazo dorado 1 y 2:	27
3.1.2.	Cálculo torque de brazo gris 1 y 2:	29
3.1.3.	Cálculo de la dimensión y motor de las ruedas.....	30
3.1.4.	Modelado 3D y simulación virtual.....	31
3.1.4.1.	Modelado e importación de robot en 3d.....	32
3.2.	Diseño electrónico.....	48
3.2.1.	Etapa de Control.....	48
3.2.2.	Diseño de la comunicación.....	50
3.2.3.	Configuración de los Módulos XBEE.....	50
3.2.4.	Configuración de cámara ip	52
3.2.5.	Etapa de Potencia.....	53
3.2.6.	Etapa de sensores y dispositivos de entrada.	55
3.3.	Interfaz de usuario	60

CAPÍTULO IV

4.	RESULTADOS Y DISCUSIÓN.....	63
4.1.	Funcionamiento mecánico	64
4.2.	Funcionamiento de la comunicación del robot.....	66
4.3.	Funcionamiento de los componentes electrónicos.	66
4.4.	Funcionamiento de la interfaz de usuario	68
4.5.	Funcionamiento del robot en campo.....	69
CONCLUSIONES		71
RECOMENDACIONES.....		72
BIBLIOGRAFÍA.....		73
ANEXOS		

ÍNDICE DE TABLAS

<i>Tabla 2-1 Principales características técnicas de la distribución de energía eléctrica nacional</i>	<i>8</i>
<i>Tabla 2-3: Principales Funciones y habilidades de robot LINESCOUT:</i>	<i>12</i>
<i>Tabla 3-1: Función API, simGetObjectHandle:</i>	<i>41</i>
<i>Tabla 3-2: Función API, simRMLMoveToJointPositions.....</i>	<i>42</i>
<i>Tabla 3-3 Tabla de conexión de dispositivos.....</i>	<i>49</i>
<i>Tabla 3-4 Tabla de características sensor IR.....</i>	<i>55</i>
<i>Tabla 3-5 Tabla de características sensor HC-SR04</i>	<i>57</i>
<i>Tabla 3-6 Características módulo GPS</i>	<i>59</i>
<i>Tabla 4-1 Análisis del robot de inspección sobre métodos tradicionales.....</i>	<i>69</i>

ÍNDICE DE FIGURAS

<i>Figura 2-1 Dispositivos en las líneas de transmisión</i>	6
<i>Figura 2-2 Dispositivos en el tendido eléctrico.</i>	7
<i>Figura 2-3 Postes de media y alta tensión.</i>	7
<i>Figura 2-4 Torres de alta tensión.</i>	8
<i>Figura 2-5 Operador inspeccionando las líneas eléctricas.</i>	10
<i>Figura 2-6: Vehículo aéreo no tripulado, Pelicano.</i>	11
<i>Figura 2-7: Robot LINESCOUT</i>	11
<i>Figura 2-8: Diseño de Robot PLIR</i>	12
<i>Figura 2-9: Estructura del robot ROBTET</i>	13
<i>Figura 2-10: Prototipo Robot ARMAN</i>	14
<i>Figura 2-11: Simulación de Snake-Like Robot</i>	14
<i>Figura 2-12: Esquema de comunicación Robot Expliner</i>	15
<i>Figura 2-13: Esquema de comunicación Inspection Robot</i>	16
<i>Figura 2-14: Modelos Xbee</i>	17
<i>Figura 2-15: Configuración de pines modulo GPS</i>	18
<i>Figura 2-16 Formato de datos GPGGA</i>	18
<i>Figura 2-17: Diagrama de conexión de pines del Arduino Mega</i>	19
<i>Figura 2-18: Espectro de Radiación vs Longitud Infrarrojo</i>	21
<i>Figura 2-19: FOV, Campo de visión sensor Infrarrojo</i>	21
<i>Figura 2-20: Bus de comunicación i2c</i>	22
<i>Figura 2-21: Trama de comunicación i2c</i>	23
<i>Figura 2-22: Formula para distancia de Sensor Ultrasónico</i>	25
<i>Figura 3-1: Características de diseño del Robot.</i>	26
<i>Figura 3-2: Principales medidas del robot</i>	27
<i>Figura 3-3: Motor para brazo dorado 1 y 2</i>	28
<i>Figura 3-4: Motor para brazo gris 1 y 2</i>	29
<i>Figura 3-5: Medidas de conductores eléctricos de media y alta tensión.</i>	30
<i>Figura 3-6: Dimensiones de rueda tipo polea con ranura.</i>	30
<i>Figura 3-7: Motor para ruedas del robot</i>	31
<i>Figura 3-8: Simulación RILE (Robot Inspector De Líneas Eléctricas)</i>	32
<i>Figura 3-9: Diseño en 3D de la estructura del robot</i>	32
<i>Figura 3-10 Exportar diseño 3D a Vrep</i>	33
<i>Figura 3-11: Diseño 3D de robot por partes.</i>	33
<i>Figura 3-12: Diseño 3D exportado en Vrep</i>	34
<i>Figura 3-13: Creación de articulaciones Vrep</i>	35

<i>Figura 3-14: Propiedades de los objetos del diseño</i>	35
<i>Figura 3-15: Objetos primitivos de geometría pura.</i>	36
<i>Figura 3-16: Creación de Objetos dinámicos y asignación de articulaciones</i>	37
<i>Figura 3-17: Asignación de jerarquías en articulaciones y objetos</i>	37
<i>Figura 3-18: Diseño final del robot configurado para simulación en Vrep</i>	38
<i>Figura 3-19: Configuración de articulaciones en modo Torque</i>	39
<i>Figura 3-20: Habilidad de motor y torque</i>	39
<i>Figura 3-21: Propiedades dinámicas del Cuerpo</i>	40
<i>Figura 3-22: Configuración de visibilidad de objetos</i>	40
<i>Figura 3-23: Creación de script de programación</i>	41
<i>Figura: 3-24 Asignación de script al Robot</i>	43
<i>Figura 3-25: Creación de Valores iniciales</i>	43
<i>Figura: 3-26 Secuencia de movimientos primera etapa</i>	44
<i>Figura: 3-27 Secuencia de movimientos 2da etapa</i>	44
<i>Figura: 3-28 Robot Rile 3D en simulación.</i>	45
<i>Figura 3-29: Secuencia de movimientos para eludir un obstáculo.</i>	46
<i>Figura: 3-30 Secuencia de movimientos para pasar un poste</i>	47
<i>Figura 3-31: Componentes del robot móvil.</i>	48
<i>Figura 3-32: Diagrama de comunicación entre el robot y el operador.</i>	50
<i>Figura 3-33: Módulo Xbee y Xbee Explorer USB</i>	51
<i>Figura 3-34: Software X-CTU para configurar módulos XBEE.</i>	51
<i>Figura 3-35: Configuración módulos XBEE.</i>	52
<i>Figura 3-36: Transmisión en directo de cámara ip.</i>	53
<i>Figura 3-37: Configuración cámara ip</i>	53
<i>Figura 3-38: Diagrama de conexiones motores y drivers</i>	54
<i>Figura 3-39: Diagrama de Flujo Control de Motores.</i>	54
<i>Figura 3-40: Diagrama de conexión sensor de temperatura.</i>	55
<i>Figura 3-41: Diagrama de Flujo Sensor de Temperatura</i>	56
<i>Figura 3-42: Diagrama de conexiones Sensores Ultrasónicos de distancia</i>	57
<i>Figura 3-43: Diagrama de flujo de Sensor de Objetos</i>	58
<i>Figura 3-44: Diagrama de conexión módulo GPS</i>	59
<i>Figura 3-45: Diagrama de los principales componentes electrónicos.</i>	60
<i>Figura 3-46: Interfaz de usuario.</i>	61
<i>Figura 3-47: Registro de alarmas de la interfaz de usuario</i>	61
<i>Figura 3-48: Registro alarmas almacenadas en Excel.</i>	62
<i>Figura 4-1: Robot de inspección sobre el cableado eléctrico.</i>	63
<i>Figura 4-2: Construcción de partes mecánicas y acoples.</i>	64
<i>Figura 4-3: Movimientos del robot para superar un obstáculo en el cableado.</i>	65

<i>Figura 4-4: Robot movilizando en cableado recto e inclinado.</i>	65
<i>Figura 4-5: Ubicación de sensores y cámara a bordo.</i>	66
<i>Figura 4-6: Gráfica de valores de temperatura del cable y ambiente.</i>	67
<i>Figura 4-7: Imagen de cámara a bordo controlada por servomotor.</i>	67
<i>Figura 4-8: Interfaz de usuario con datos de funcionamiento.</i>	68
<i>Figura 4-9: Simulación y funcionamiento del robot.</i>	69

RESUMEN

Se diseñó e implementó un robot móvil para la inspección de líneas eléctricas, que busca investigar alternativas que tengan las principales características necesarias para realizar estas tareas de inspección, e implementarlas en un robot de costos accesibles. En el diseño del robot se modelaron sus partes mecánicas en 3D y posteriormente llevarlo a un software de simulación virtual (Vrep), en el cual se programó las secuencias de movimientos.

La implementación y desarrollo del robot, permite medir la temperatura del cable y detectar objetos. Además, consta de una cámara que transmite en tiempo real el desplazamiento del robot y su entorno. El operador controla de forma remota el robot mediante una interfaz de usuario.

Más de 30 ensayos se realizaron en el complejo de pruebas de la EEASA, donde se comprobó su correcto funcionamiento, con lo que se concluye que se disminuye el riesgo de realizar las inspecciones en un 100%, ya que el operador no está en contacto con el tendido eléctrico, además de disminuir la cantidad de trabajo realizado a un 33%, ya que se realizan 3 inspecciones simultáneamente. Se recomienda el uso del robot en climas favorables, pero no realizar trabajos en lluvias.

Palabras Claves:

<TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA> <INGENIERÍA ELÉCTRICA, ROBÓTICA> <INSPECCIÓN DE LÍNEAS ELÉCTRICAS> <ROBOT DE INSPECCIONES> <COMUNICACIONES XBEE> < ARDUINO (SOFTWARE - HARDWARE)> <LABVIEW (SOFTWARE)>

SUMMARY

A mobile robot was designed and implemented for the inspection of electrical lines, which seeks to investigate alternatives that have the main necessary characteristics to perform these inspection tasks and implement them in a robot of accessible costs. In the robot design was modeled its mechanical parts in 3D and then previously taken to a virtual simulation software (Vrep), in which the sequence of movements were programmed.

The implementation and development of the robot allows to measure the temperature of the cable and to detect objects. In addition, it consists of a camera that transmits in real time the displacement of the robot and its environment. The operator controls in a remote mode the robot through a user interface.

More than 30 tests were carried out in the EEASA test complex, where they were checked for correct operation thus it is concluded that the risk of performing inspections is diminished by 100%, since the operator is not in contact with the power grids. In addition to diminishing the amount of work carried out by 33%, since 3 inspections are carried out simultaneously. The research paper recommends to use the robot in favorable climates, but not work in rain.

Palabras Claves:

<TECHNOLOGY AND SCIENCES OF ENGINEERING> <ELECTRICAL ENGINEERING, ROBOTICS> <INSPECTION OF ELECTRICAL LINES> <ROBOT OF INSPECTIONS> <COMMUNICATIONS XBEE> < ARDUINO (SOFTWARE-HARDWARE)> <LABVIEW (SOFTWARE) >

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Título

Diseño e implementación de un robot móvil, para la inspección de líneas eléctricas.

1.2. Planteamiento de problema

La dificultad en la inspección de líneas eléctricas, manifestado por el difícil acceso a las líneas eléctricas o por la gran extensión del tendido eléctrico, puede llevar a que no se realice un mantenimiento preventivo en el tiempo correcto. El alto riesgo que conlleva el acceder a las alturas en las que se encuentran instaladas las líneas eléctricas, hacen que se realicen con mínima frecuencia, lo cual no permitirá una pronta detección de posibles daños o fallos en el cableado eléctrico lo que a futuro conllevaría a problemas y hasta suspensión de la distribución de energía.

La dificultad en la inspección de líneas eléctricas y la falta de mantenimientos preventivos y correctivos, determina la necesidad de definir nuevos métodos de inspección con el uso de la robótica y varias tecnologías, que faciliten y optimicen la ejecución de esta tarea, además de disminuir el riesgo al realizarla

1.3. Formulación del problema

¿Un robot móvil puede optimizar las tareas y disminuir el riesgo de inspección de las líneas eléctricas, reemplazando los métodos actualmente realizado por operarios?

1.4. Sistematización del problema

¿Cuáles son los métodos actuales para realizar las tareas de inspección en el tendido eléctrico?

¿Cómo se puede detectar una posible avería o necesidad de mantenimiento en las líneas eléctricas?

¿Cuáles son los tipos de inspecciones y mantenimientos realizados en el tendido eléctrico?

¿Qué problemas pueden producirse debido a la falta de inspección y mantenimiento de las líneas eléctricas?

¿Qué tareas implicarían un mantenimiento correctivo, al no poder detectar anticipadamente un problema en las líneas de alta tensión?

¿Qué tiempo involucra inspeccionar las líneas eléctricas con métodos convencionales?

1.5. Objetivos de la investigación

1.5.1. Objetivo general

Diseñar y desarrollar un robot móvil para optimizar los métodos de inspección de las líneas eléctricas.

1.5.2. Objetivos específicos.

- Analizar las estrategias aplicadas en la actualidad para la inspección de líneas de transmisión eléctrica con robots móviles.
- Diseñar la estructura del robot móvil para desplazarse sobre las líneas eléctricas.
- Desarrollar un sistema de comunicación inalámbrico que permita transmitir la información adquirida por el robot móvil.
- Diseñar una interface gráfica de usuario para el control del robot móvil desde un ordenador.
- Desarrollar un sistema para la detección de objetos que permita almacenar su posición mediante odometría o posicionamiento global.
- Transmitir video en tiempo real, del entorno del robot móvil mientras se desplaza en las líneas eléctricas.

1.6. Justificación de la investigación

En la actual sociedad el uso de la energía eléctrica es una de las principales fuentes de energía, la misma que necesita ser trasladada desde su lugar de generación como centrales hidroeléctricas, térmicas, etc., hasta las subestaciones y posterior uso en residencias o industrias. En medio de este proceso, se necesita conducir energía por medio de tendidos eléctricos, los cuales, con el tiempo, deben tener un mantenimiento preventivo para poder estar en correcto funcionamiento.

Antes de entrar en un proceso de mantenimiento es importante saber el estado en el que se encuentran los cables del tendido eléctrico, sus accesorios y su entorno. Estas tareas se desarrollan por medio de inspecciones visuales realizadas por las personas encargadas, lo que conlleva una gran dificultad de acceso al operario de mantenimiento, e indiscutiblemente varios riesgos en la realización de la inspección, debido a la gran extensión del tendido eléctrico y al difícil acceso por la situación geográfica en donde se encuentran instaladas las torres y tendido eléctrico, debido a montañas, quebradas, etc.

En la actualidad existen varios métodos para realizar la inspección de las líneas eléctricas, e inclusive para realizar el mantenimiento de las mismas, pero aún tienen un alto costo para llevarlas a cabo o para acceder a ese tipo de servicio.

Por lo que con el presente proyecto se busca investigar alternativas que tengan las principales características que se necesitan para realizar estas tareas de inspección, y poder implementarlas en un robot de costos accesibles para poder dar este tipo de servicios, optimizando y disminuyendo el riesgo y tiempo. Se plantea que, con la implementación del robot móvil en las tareas de inspección, primero se optimice dicha tarea, en tiempo, precisión y varios datos adquiridos en una sola inspección, segundo se disminuya el riesgo que conlleva el realizar varias de estas tareas por un operario, y finalmente obtener una base de datos a partir de los historiales de las inspecciones que se puedan realizar con el robot móvil.

1.7. Hipótesis

Al implementar un robot móvil para la inspección de líneas eléctricas, se optimiza las tareas de inspección, disminuyendo el tiempo al obtener varios datos del tendido eléctrico y su entorno, además de disminuir el riesgo que conlleva realizar estas tareas.

CAPÍTULO II

2. MARCO DE REFERENCIA

Como consumidores de energía eléctrica cada vez se es más consciente de la importancia del comportamiento y hábitos para lograr avances en la eficiencia energética, pero el consumidor es el último eslabón de una larga cadena que se inicia en el punto de generación de la energía y hasta llegar al usuario final a través de la red de transporte. La eficacia del sistema eléctrico depende de su correcto funcionamiento, minimizando las pérdidas en el recorrido y adecuando la oferta, así como la demanda en tiempo real.

2.1. Estructuras de las líneas eléctricas

Un sistema eléctrico puede transportarse como energía eléctrica en corriente continua y corriente alterna, el 99% de los suministros se realizan mediante corriente alterna trifásica.

El transporte de corriente eléctrica desde donde se genera hasta donde se consume, conlleva pérdidas magnéticas, por lo que se requiere disminuir la intensidad que circula por las mismas, este hecho obliga a realizar el transporte de la corriente a una tensión elevada.

Una vez en el lugar de consumo, se reduce la tensión, hasta alcanzar valores estándares que se utilizan a nivel industrial y doméstico. Para esto se emplean los transformadores, para disminuir o elevar la tensión, conservando la misma frecuencia.

Los generadores de corriente alterna instalados en centrales eléctricas suelen producir tensión de hasta 25Kv. Esta tensión no es lo suficientemente elevada para el transporte de grandes potencias, por lo que se eleva, mediante transformadores, la tensión hasta alcanzar valores de cientos de miles de voltios, con lo que es posible el transporte de grandes potencias con pequeñas intensidades, lo que disminuye las pérdidas, posteriormente en el lugar de consumo, se reduce la tensión, hasta alcanzar valores estándares. (Checa, 2006)

2.2. Inspección y mantenimiento de líneas eléctricas

Las líneas eléctricas son un entorno complejo y peligroso, con intensos campos eléctricos y magnéticos, lo que dificulta el movimiento de los robots.

En la Figura 2-1 se muestran varios dispositivos utilizados en las líneas de transmisión que serán obstáculos que podrán interferir en el recorrido del robot. Las líneas de transmisión más simples tienen un conductor por fase colgado en las cuerdas aislantes (3), que pueden ser suspensores (4) o aisladores de tensión. También hay amortiguadores de vibraciones (5), luces de advertencia de aviones (7) y abrazaderas. En las líneas eléctricas, que tienen más de un conductor por fase (1), hay incluso más obstáculos tales como espaciadores y amortiguadores espaciadores. (Gonçalves, 2013)

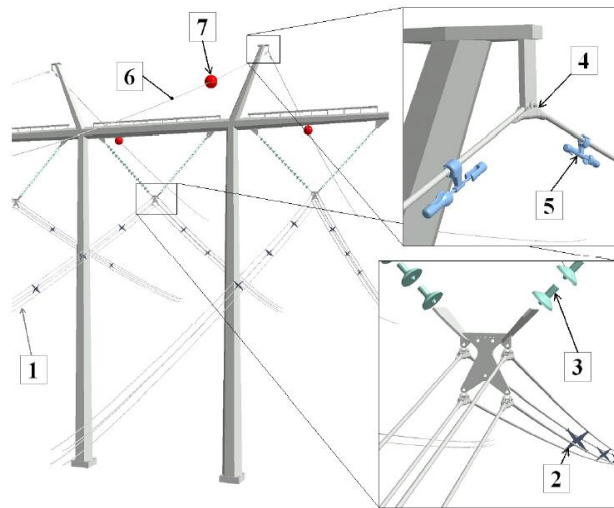


Figura 2-1 Dispositivos en las líneas de transmisión

Fuente: Checa, 2006

Tres obstáculos típicos pueden ser identificados en las líneas eléctricas que un robot móvil tendrá que superar cuando se utiliza para reparar y / o inspeccionar las líneas. El primero es el contrapeso (o amortiguador de vibraciones) que es el más común de los obstáculos. El segundo tipo de obstáculo es la esfera de advertencia del avión (o esfera de señalización) y el tercero es la torre que sostiene el alambre.

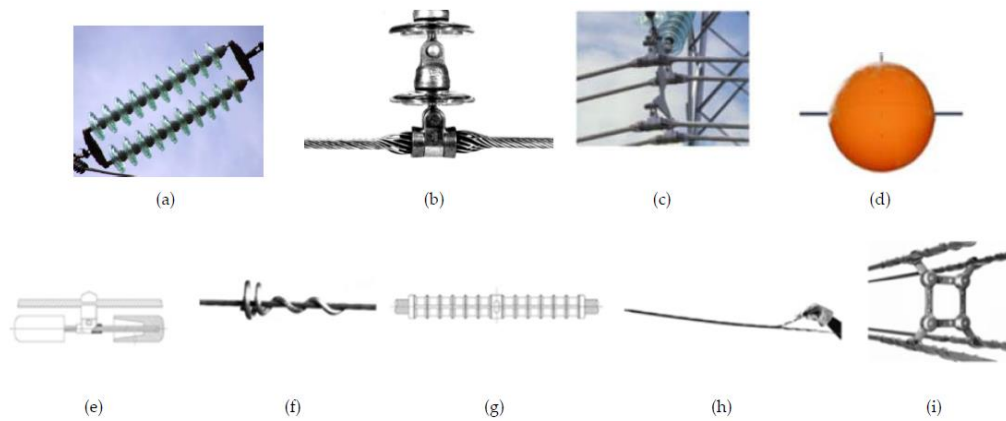


Figura 2-2 Dispositivos en el tendido eléctrico.

Fuente: Gonçalves, 2013

Estos aparatos son inspeccionados por los trabajadores utilizando binoculares que se mueven a lo largo de las líneas de transmisión. Generalmente, los procedimientos de mantenimiento se realizan cuando el problema ya ha ocurrido (mantenimiento correctivo).

En la Figura 2-3 se muestran ejemplos de postes para las líneas eléctricas y que el robot móvil deberá cruzarlas.

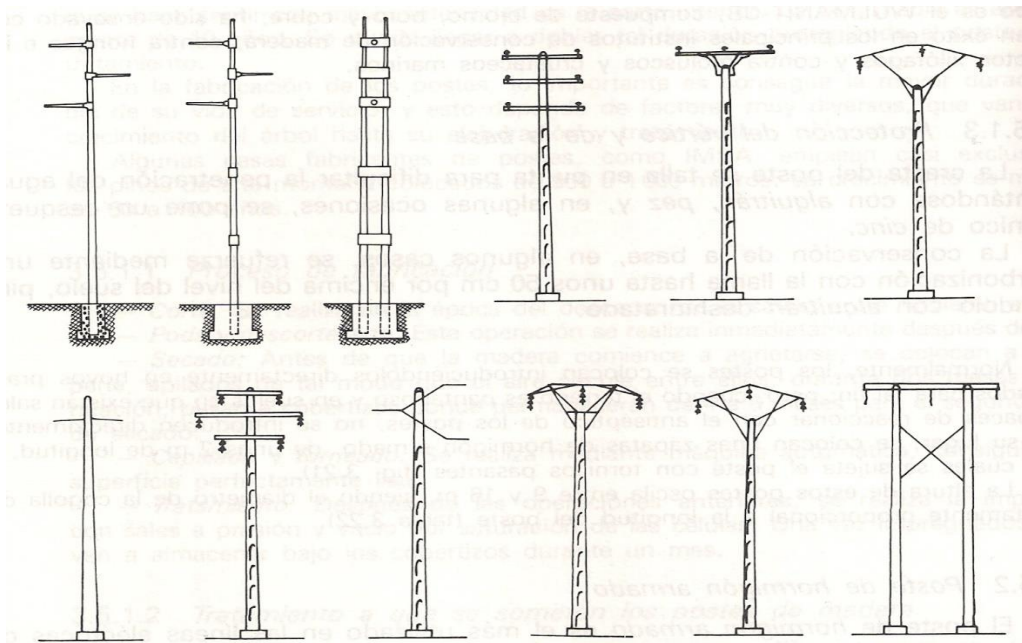


Figura 2-3 Postes de media y alta tensión.

Fuente: Gonçalves, 2013

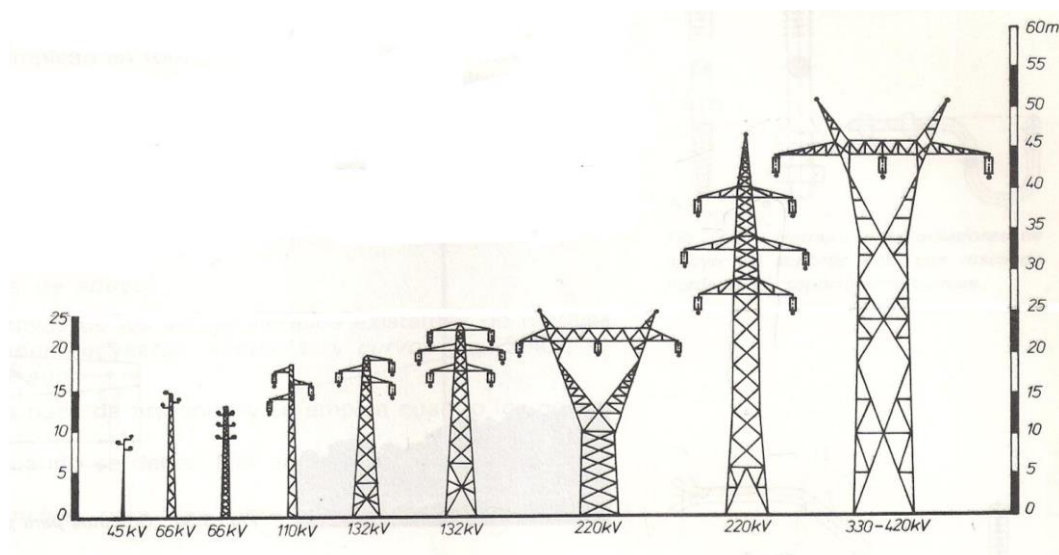


Figura 2-4 Torres de alta tensión.

Fuente: Gonçalves, 2013

2.3. Características de las líneas de distribución de energía en el Ecuador

Las líneas de distribución en el país, tienen características que se deben tomar en cuenta, como por ejemplo el total de las líneas de transmisión y subtransmisión es de 4558 Km, además de 66630 Km del tendido eléctrico de las redes de media tensión, es importante destacar estos valores, para poder estimar la dificultad que sería poder inspeccionar esta cantidad de líneas eléctricas.

Tabla 2-1 Principales características técnicas de la distribución de energía eléctrica nacional

Grupo	Empresa	Centrales de Generación			Subestaciones de Distribución (MVA)	Demanda Máxima del Sistema (MW)	Líneas de transmisión y subtransmisión (km)	Redes de Medio Voltaje (km)	Transformadores de Distribución			Redes de Bajo Voltaje (km)	Luminarias		Acometidas (#)	Medidores (#)
		Potencia Nominal (MW)	Potencia Efectiva (MW)	Cantidad N°					Monofásico (#)	Trifásico (#)	Total (MVA)		Cantidad (#)	Potencia (kW)		
Corporación Nacional de Electricidad CNEL	CNEL-Bolívar	1,66	1,33	1	23	15	109	745	-	-	-	15.667	10.479	2.427	52.410	52.044
	CNEL-EI Oro	0,27	0,22	2	196	114	259	3.459	7.638	1.271	263	49.624	62.454	11.069	193.075	186.158
	CNEL-Esmeraldas	1,10	0,90	1	144	80	266	2.307	4.320	581	167	16.716	25.731	5.188	101.239	92.386
	CNEL-Guayas Los Ríos				283	62	380	1.820	15.760	1.163	417	65.297	49.548	10.249	257.071	235.462
	CNEL-Los Ríos				65	223	98	1.586	5.123	318	130	18.348	13.548	3.053	88.245	88.245
	CNEL-Manabí				313	130	722	6.389	19.638	454	547	166.760	92.220	21.060	255.229	247.519
	CNEL-Miagro				173	72	231	2.123	6.239	286	164	11.469	31.476	5.769	126.088	126.088
	CNEL-Sta. Elena				116	71	184	1.425	5.201	134	165	17.007	28.679	4.839	101.851	101.851
	CNEL-Sto. Domingo				107	38	160	4.218	10.458	550	219	18.099	33.340	4.888	115.549	143.683
Total CNEL	CNEL-Sucumbios	43,64	32,04	7	120	239	133	2.569	3.307	408	87	28.411	18.438	2.344	62.380	61.318
		46,66	34,49	11	1.540	1.043	2.562	28.640	77.885	5.165	2.181	407.597	365.913	70.884	1.353.137	1.334.754
Empresas Eléctricas	E.E. Ambato	8,00	6,20	2	195	91	119	4.138	9.207	1.715	259	72.862	51.015	7.778	211.144	211.144
	E.E. Azogues				13	16	27	627	1.172	150	27	12.765	9.979	1.458	30.796	30.797
	E.E. Centro Sur	0,50	0,40	1	261	141	274	7.362	12.472	2.952	445	118.223	78.537	12.951	246.412	299.507
	E.E. Cotacachi	12,19	11,88	5	115	67	116	2.733	4.303	624	108	50.464	27.714	4.116	101.346	101.544
	E.E. Galapagos	9,81	7,69	5	14	6	12	174	352	88	13	2.478	2.234	248	6.777	6.364
	E.E. Norte	14,77	14,07	4	198	87	243	4.846	10.849	1.989	318	60.215	60.433	7.591	137.215	195.110
	E.E. Quito	140,37	136,05	8	1.456	647	268	7.113	19.039	13.284	2.041	76.420	198.911	29.374	418.078	849.325
	E.E. Riobamba	16,34	15,30	4	116	52	155	3.181	7.788	497	153	59.782	25.199	3.617	140.604	145.638
	E.E. Sur	16,31	14,40	2	110	51	554	6.644	11.463	405	172	43.544	39.164	4.797	111.058	155.238
	Eléctrica de Guayaquil	236,07	212,00	3	1.050	783	229	1.142	23.039	915	1.491	35.750	134.583	20.763	498.061	571.964
	Total Empresas Eléctricas	454,37	417,99	34	3.485	1.940	1.996	37.990	99.684	22.599	5.027	532.504	627.769	92.700	1.901.491	2.568.531
TOTAL NACIONAL		501,03	452,48	45	5.025	2.984	4.558	66.630	177.569	27.764	7.207	940.101,27	993.682	163.584	3.254.628	3.903.285
Empresa no presenta información																

Fuente: CONELEC -Informe estadístico sector eléctrico ecuatoriano 2010

2.4. Procedimiento manual de inspección y mantenimiento de líneas eléctricas

A continuación, se explicará el procedimiento y equipamiento que se utiliza actualmente para la inspección y mantenimiento de las líneas eléctricas de alta tensión, a fin de poder comparar las ventajas que ofrece el uso de un sistema robotizado para la realización de esta tarea. (León Rivas, 2013)

Equipo: En primera instancia, el operario que realiza la inspección estará dotado con un equipo de protección personal para evitar los posibles accidentes derivados de la manipulación de elementos de alta y media tensión. Para ello contará, entre otros de los siguientes elementos:

- Guantes aislantes, fundas, mantas cubiertas y estereras: necesarias para evitar el contacto de forma inadvertida con los cables de alta tensión:
- Casco: absorbe la mayor parte de la electricidad en caso de contacto con la línea.
- Gafas: proteger los ojos de las chispas.

Herramientas de manipulación: Los operarios para tocar los elementos conductivos de la red utilizan una herramienta llamada pértiga. La mayoría de estas herramientas se construyen a partir de fibra de vidrio con aislamiento de alta tensión que envuelve la empuñadura del mango. Además, se ajustan a diferentes longitudes con un diseño telescópico de forma que permite manipular o manejar cables de alta tensión a los operarios de forma flexible.

Equipamiento de escalada: Para acceder a los postes de alta tensión los operarios disponen de escaladores y cintas de postes que les permiten escalar los postes de madera. Estas cintas se envuelven alrededor del poste y se conectan a un cinturón de escalada alrededor de su cintura. Los escaladores, además, poseen puntos o controladores en la parte inferior de las polainas que golpea el poste de madera y mantienen al operario en su lugar mientras avanza. Las cintas de poste, por el contrario, se desliza hacia arriba sobre la parte exterior del poste, mientras sube el operario, sosteniéndolo en su lugar cuando se incline hacia atrás y apriete la cinta alrededor del poste. (Armendáriz, 2016)



Figura 2-5 Operador inspeccionando las líneas eléctricas.

Fuente: Armendáriz, 2015.

Las líneas de transmisión están expuestas a una variedad de factores, tales como la corrosión y las vibraciones inducidas por el viento, que causan problemas diferentes y limitan la vida útil de las líneas.

Por ejemplo, el Sector Eléctrico Brasileño tiene una red de transmisión de más de 100.000 Km.

Líneas de transmisión con voltajes superiores a 230 kV que necesitan inspección y mantenimiento continuos. (Elizondo, Gentile, & Candia, 2010)

2.5. Robots existentes para inspección de líneas eléctricas

En este apartado se mostrarán varios de los robots y proyectos existentes que se han llevado a cabo para la inspección de líneas eléctricas.

2.5.1. Proyecto pelicano

Proyecto finalizado en el año 2010 llevado a cabo por Red Eléctrica de España y el grupo de Medida, Control y Comunicaciones de AIN, el cual desarrolló un sistema de robot aéreo, tipo helicóptero para su aplicación a la inspección de líneas eléctricas. Se trata de un vehículo aéreo no tripulado capaz de navegar de forma autónoma y capturar imágenes de calidad equivalente a las obtenidas, actualmente, mediante la inspección intensiva con helicópteros tripulados. La nueva metodología pretende minimizar los riesgos de seguridad y coste comparado con la práctica actual. (AIN, REE, 2004)



Figura 2-6: Vehículo aéreo no tripulado, Pelicano.

Fuente: AIN, REE, 2004

2.5.2. Linescout

Se trata de un robot capaz de inspeccionar líneas eléctricas además de evitar los obstáculos que se pudieran encontrar en estas. Este robot, además, es capaz de transmitir y recibir datos y video a través de dos radiofrecuencias con un radio de cinco Km de distancia. Para ello está dotado de tres cámaras de video situadas en la base, las cuales son utilizadas como realimentación del control del robot para dirigir el movimiento de los diferentes ejes que lo componen. (Nicolas Pouliot, 2015)



Figura 2-7: Robot LINESCOUT

Fuente: Nicolas Pouliot, 2015

Tabla 2-2: Principales Funciones y habilidades de robot LINESCOUT:

1	Inspecciones visuales por cuatro cámaras de operador.
2	Los componentes defectuosos de la línea de transmisión pueden ser identificados ya sea a través de cámaras clásicas o tecnología infrarroja
3	Reparación temporal de componentes. Utilizando una llave de torsión eléctrica universal, los componentes podrían ser reajustados. Además, los cables de conductor rotos pueden repararse temporalmente instalando una abrazadera
4	Aplicación de sensores como la medición de empalmes de compresión

Fuente: Overview of Robotic Applications for Transmission Line Work, 2015

2.5.3. Robot PLIR

En este caso se muestra una solución propuesta para la superación de obstáculos para robots de inspección de líneas de alta tensión. Esta configuración permite salvar tanto obstáculos conocidos como desconocidos. Para obstáculos ya conocidos (clasificados mediante un sistema de procesamiento de imágenes), trayectorias precalculadas serán ejecutadas para sortearlo. Para desconocidas, sin embargo, el PLIR desarrollará un conjunto de maniobras online para su solución. (Rogério Sales Gonçalves, 2013)

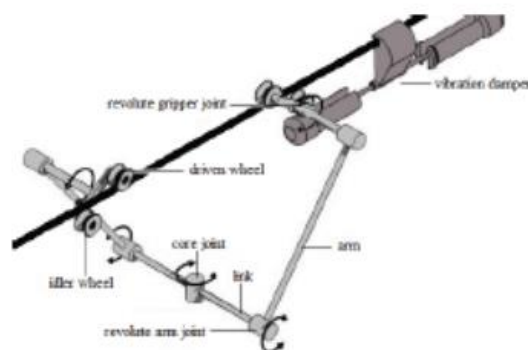


Figura 2-8: Diseño de Robot PLIR

Fuente: Rogério Sales Gonçalves, 2013

2.5.4. Robtet

Se trata de un sistema telerrobótico y semiautomático desarrollado por la Universidad Politécnica de Madrid para el mantenimiento de las líneas de distribución eléctrica. Fue concebido bajo la iniciativa intergubernamental de apoyo a la I+D+i EUREKA y ha sido uno de los mayores éxitos en el campo de la robótica y la producción automática. ROBTET se compone de dos partes, un mástil auxiliar y varias videocámaras estereoscópicas, todo ello en conjunto montado sobre una plataforma en el extremo de la pluma de un camión-grúa guiado y monitorizados desde la cabina de control. (R. Aracil, 1995)

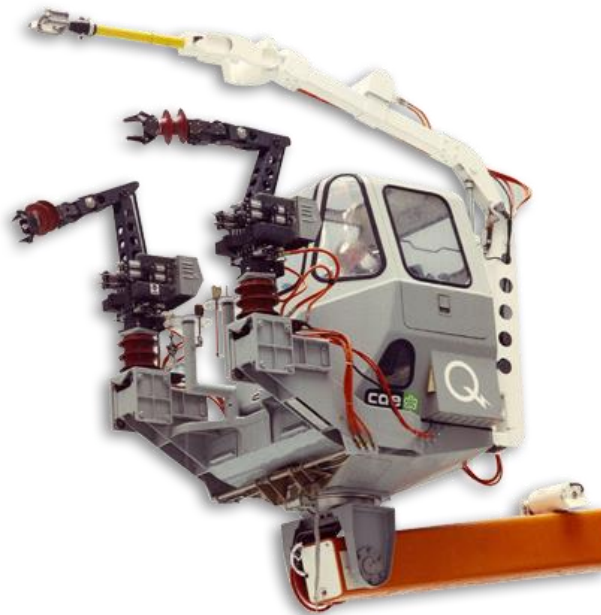


Figura 2-9: Estructura del robot ROBTET

Fuente: R. Aracil, 1995

2.5.5. Arman

Se trata de un robot autosustentable desarrollado por el instituto de investigación en Electrónica de Potencia (EPRI) de Estados Unidos, capaz de recorrer las líneas eléctricas para labores de inspección, análisis y estudio en áreas de difícil acceso. Su capacidad es de unos 130 km y se puede realizar un par de veces por año. Este desplazamiento se produce además mediante movimientos deslizantes a través de los cuales irá recorriendo las líneas del tendido eléctrico. (Barajas, 2010)



Figura 2-10: Prototipo Robot ARMAN

Fuente: Barajas, 2010

2.5.6. *Snake-like*

El robot Snake-Like, lleva su nombre en función a que su mecanismo de locomoción se basa en las culebras. Tiene la capacidad de recorrer y escalar líneas eléctricas, lo que permite evitar los obstáculos que se encuentran en el recorrido. Las juntas del mecanismo están compuestas de dos espacios ortogonales lo que permite atravesar los obstáculos en el recorrido de las líneas de transmisión, para ello se realizó el estudio de un algoritmo para el movimiento, con la cinemática y dinámica de los eslabones que componen el robot. (Wang Wei, 2013)

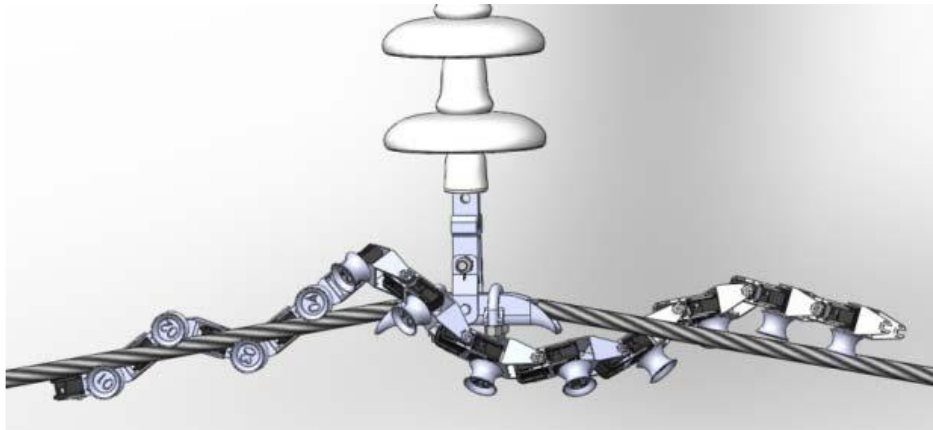


Figura 2-11: Simulación de Snake-Like Robot

Fuente: Wang Wei, 2013

2.6. Tipos de comunicación de robot de inspección de líneas eléctricas

La comunicación entre el operador y el robot, en todos los casos se realiza de manera inalámbrica. La comunicación inalámbrica se realizó por vía Inalámbrica TCP-IP, por medio de comunicación de radio frecuencia o por comunicación mediante Xbee. Como en los siguientes casos:

Robot Expliner, La comunicación se basa en wireless Lan TCP/IP, La conexión Wifi se encuentra entre la caja de control y el CPU del robot. (Debenest, 2008)

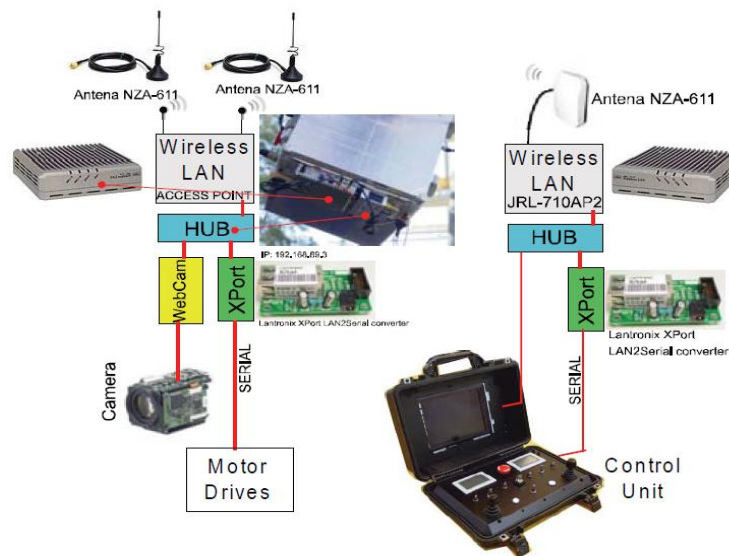


Figura 2-12: Esquema de comunicación Robot Expliner

Fuente: Debenest, 2008

Inspection Robot, La comunicación entre el lugar de control y el robot se realiza mediante Wifi 802.11, pero además existe comunicación Xbee para el control de las ruedas del robot. (Debenest, 2008)

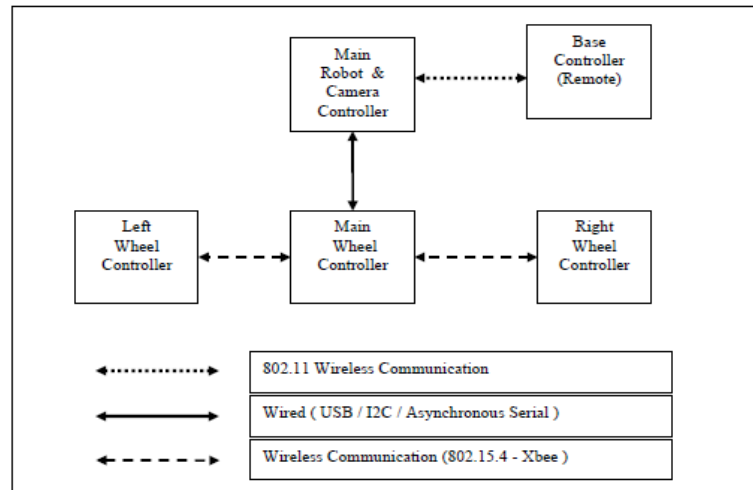


Figura 2-13: Esquema de comunicación Inspection Robot

Fuente: Debenest, 2008

2.7. Módulos de comunicación xbee

Los módulos XBEE, son soluciones integradas de tipo chip, que brindan comunicación inalámbrica para interconectar dos o más dispositivos utilizando el protocolo IEEE 802.15.4 puede ser punto a punto o punto a multipunto. Pueden ser fácilmente reemplazados en lugar de un par de cables en una comunicación serial. Xbee es propiedad de Digi la misma que se basa en el protocolo Zigbee.

Zigbee es una alianza y un estándar de redes MESH de gran eficiencia energética y bajos costes.

Tipos de Antena:

Chip Antenna: Básicamente es un pequeño chip que actúa como antena. Rápido, sencillo y barato.

Wire Antenna: (Whip Antenna) Es un pequeño cable que sobresale.

u.FL Antenna: Un conector pequeño para conectar tu propia antena. Esto es perfecto si tienes tu equipo en una caja y deseas la antena afuera de ésta.

RPSMA Antenna: Un conector más grande para conectar tu propia antena. Nuevamente, esto es perfecto si tienes tu equipo en una caja y deseas la antena afuera de ésta.

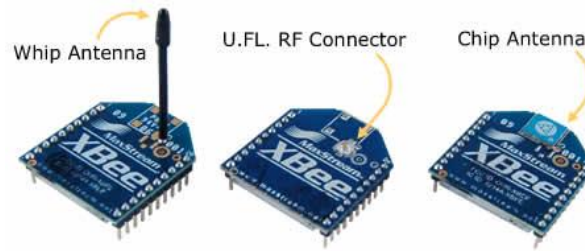


Figura 2-14: Modelos Xbee

Fuente: MCI electronics, 2011

Además de la versión regular o normal de los módulos Xbee, existe también una versión PRO, la diferencia entre ambos a nivel de hardware es principalmente el rango de alcance. La versión Pro tiene un mayor alcance (1,6 Km línea vista), pero para ello tiene un mayor consumo de potencia. (MCI electronics, 2016).

2.8. Módulo gps

El módulo GPS trabaja con un estándar, raw NMEA0183 (National Marine Electronics Association) o también con una solicitud de información requerida mediante interface serial, con un seguimiento superior de hasta 20 satélites.

El modulo provee la fecha actual, tiempo, latitud, longitud, altitud, velocidad, y dirección de recorrido.

Las áreas de aplicación del dispositivo son: navegación, sistemas de seguimiento, mapeo, gestión de flotas, piloto automático y robótica. (Parallax Inc., 2011)

Características:

- Integración completa.
- GPS de bajo costo.
- Antena pasiva.
- Comunicación TTL.

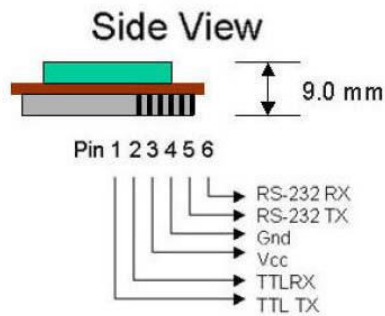


Figura 2-15: Configuración de pines modulo GPS

Fuente: Parallax Inc., 2011

El modulo permite trabajar en formato RAW el cual tiene el NMEA0183 que proporciona una serie de cadenas ASCII delimitadas por comas, cada una precedida de un identificador en el encabezado. Los datos se transmiten como una cadena de 4800bps de caracteres ASCII de 8 bits. Así, cualquier microcontrolador con un puerto serie puede extraer datos del módulo GPS. Pero, los módulos no producen Información de ubicación de "texto sin formato". En su lugar, crean "oraciones" estandarizadas, tales como (SiRF Technology, 2005):

\$ GPGGA, 170834,4124,8963, N, 08151,6838, W, 1,05,1,5,280.2, M, -34,0

\$ GPGSA, A, 3,19,28,14,18,27,22,31,39 ,,,, 1,7,1,0,1,3 * 34

\$ GPGSV, 3,2,11,14,25,170,00,16,57,208,39,18,67,296,40,19,40,246,00 * 74

\$ GPRMC, 220516, A, 5133,82, N, 00042,24, W, 173,8,231,8,130694,004,2

Name	Example	Units	Description
Message ID	\$GPGGA		GGA protocol header
UTC Time	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-4
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	9.0	meters	
Units	M	meters	
Geoid Separation		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR> <LF>			End of message termination

Figura 2-16 Formato de datos GPGGA

Fuente: Parallax Inc., 2011

2.9. Arduino mega 2560

Arduino es una tarjeta electrónica basada en microcontrolador AtMega, que al ser un dispositivo Open Source, ofrece muchas facilidades en proyectos electrónicos de pequeña, media y hasta gran envergadura. Arduino puede utilizarse en el desarrollo de proyectos interactivos autónomos o puede comunicarse a un PC a través del puerto serial (conversión con USB) utilizando lenguajes como Flash, Processing, MaxMSP, etc. (Arduino, 2017)

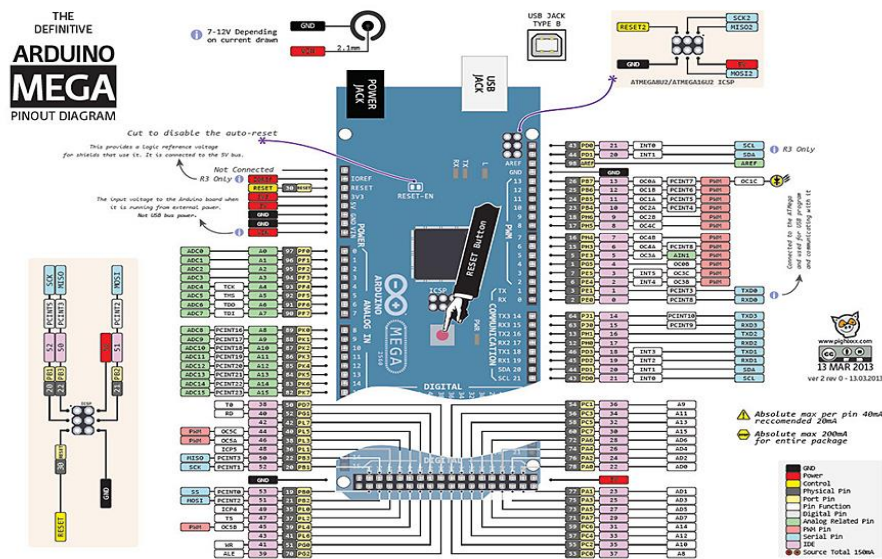


Figura 2-17: Diagrama de conexión de pines del Arduino Mega

Fuente: Arduino, 2017

Arduino Mega posee las siguientes especificaciones:

- Microcontrolador: ATmega2560
- Voltaje Operativo: 5V
- Voltaje de Entrada: 7-12V
- Voltaje de Entrada(límites): 6-20V
- Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)
- Pines analógicos de entrada: 16
- Corriente DC por cada Pin Entrada/Salida: 40 mA

- Corriente DC entregada en el Pin 3.3V: 50 mA
- Memoria Flash: 256 KB (8KB usados por el bootloader)
- SRAM: 8KB
- EEPROM: 4KB
- Clock Speed: 16 MHz

2.10. Sensor infrarrojo de temperatura

El Sensor MLX90614 es un sensor de temperatura infrarrojo “SIN CONTACTO”. Es posible conectar este sensor con un autómatas o procesador como Arduino para medir la temperatura de un objeto a distancia.

Existen distintos modelos del MLX90614 cada uno con un sufijo de tres letras. Los diferentes sensores difieren en el voltaje de operación, el número de sensores infrarrojos, y la posición del filtro.

La comunicación se realiza a través de SMBus, un subconjunto de bus I2C, por lo que resulta sencilla su lectura, y es posible conectar más de un sensor de forma simultánea.

Frecuentemente se encuentran termómetros MLX90614 integrados en módulos como la GY-906 que incorporan la electrónica necesaria para conectarla de forma sencilla a un Arduino. En la mayoría de los módulos, esto incluye un regulador de voltaje que permite alimentar directamente a 5V.

Este tipo de termómetros infrarrojos tienen un gran número de aplicaciones, incluyendo sistemas de control de temperatura en instalaciones térmicas en edificios, control industrial de temperatura, detección de movimiento, y aplicaciones de salud, emulando una cámara infrarroja. (Melexis, 2009)

2.10.1. Funcionamiento sensor MLX90614

Según la ley de Stefan-Boltzmann, todo objeto por encima del cero absoluto (°K) emite radiación cuyo espectro es proporcional a su temperatura. El MLX90614 recoge esta radiación y su salida es una señal eléctrica proporcional a la temperatura de todos los objetos en su campo de visión.

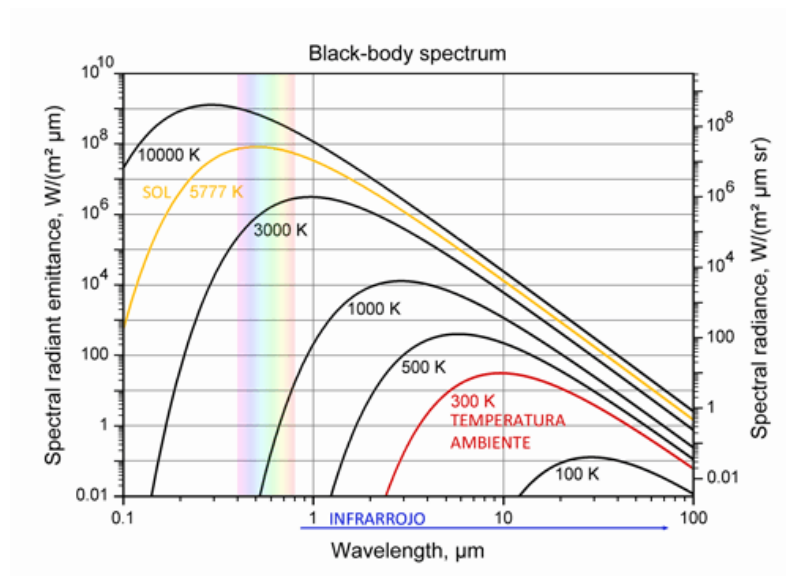


Figura 2-18: Espectro de Radiación vs Longitud Infrarrojo

Fuente: Melexis, 2009

Internamente el MLX90614 está constituido con un chip de silicio con una fina membrana micromecanizada sensible a la radiación infrarroja, junto con la electrónica necesaria para amplificar y digitalizar la señal y calcular la temperatura. Viene calibrado de fábrica en un amplio rango de temperaturas: -40 a 85 °C para la temperatura ambiente y -70 a 382 °C para la temperatura de objetos. La precisión estándar es de 0.5 °C referente a la temperatura ambiente.

El MLX90614 dispone de dos modos de salida. La estándar es SMBus, un conjunto del I2C, con una resolución de 0.02°C. También puede emplear una salida PWM de 10 bits para mediciones continuas, aunque con menor resolución 0.14°C.

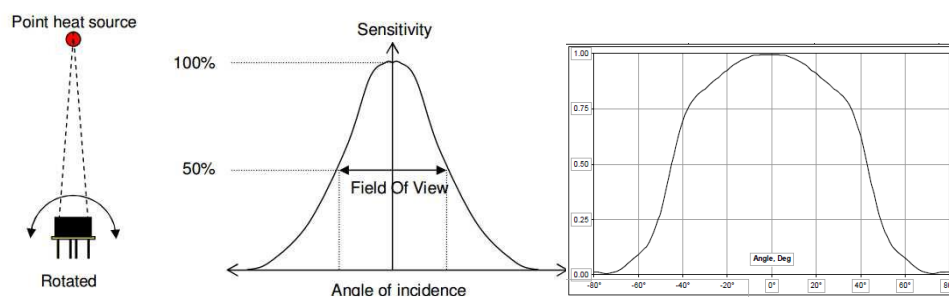


Figura 2-19: FOV, Campo de visión sensor Infrarrojo

Fuente: Melexis, 2009

También es importante entender que el MLX90614 es sensible a todos los objetos ubicados en su campo de visión. El ángulo de visión depende del modelo, y varía desde 5° a 80°C. En el ángulo más amplio de 80°, el área de medición a 0.5 tiene un diámetro de 0.83 metros

Es decir, los modelos de menos ángulo son apropiados para medidas puntuales en frente del sensor. Los sensores de ángulo amplio están diseñados para detectar incrementos de temperatura en una gran zona, por ejemplo, para detección de fallas en maquinaria. (Melexis, 2009)

2.11. Comunicación i2c

El estándar I2C (Inter-Integrated Circuit) fue desarrollado por Philips en 1982, para la comunicación interna de dispositivos electrónicos en sus artículos. Posteriormente fue adoptado progresivamente por otros fabricantes hasta convertirse en un estándar del mercado. I2C también se denomina TWI (Two Wired Interface) ya que requiere únicamente dos cables para su funcionamiento, uno para la señal de reloj (CLK) y otro para el envío de datos (SDA), lo cual es una ventaja frente al bus SPI. Por contra, su funcionamiento es un poco más complejo, así como la electrónica necesaria para implementarla. (Prometec, 2017)

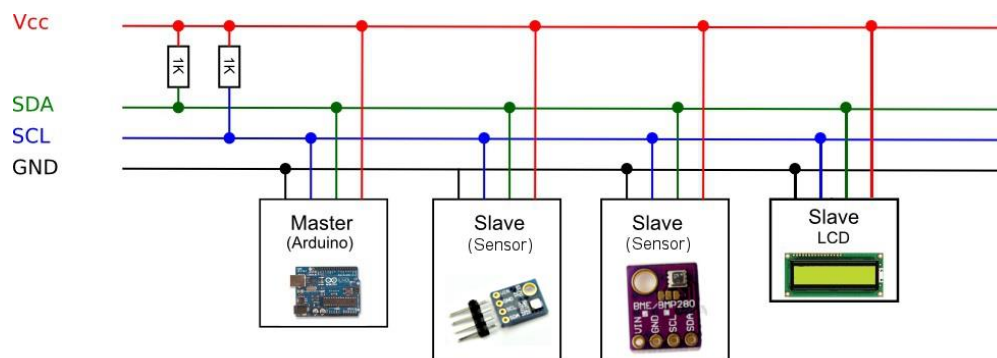


Figura 2-20: Bus de comunicación i2c

Fuente: Prometec, 2017

En el bus cada dispositivo dispone de una dirección, que se emplea para acceder a los dispositivos de forma individual. Esta dirección puede ser fijada por hardware o totalmente por software.

El bus I2C tiene una arquitectura de tipo maestro-esclavo. El dispositivo maestro inicia la comunicación con los esclavos, y puede mandar o recibir datos de los esclavos. Los esclavos no pueden iniciar la comunicación (el maestro tiene que preguntarles), ni hablar entre si directamente.

El bus I2C es síncrono. El maestro proporciona una señal de reloj, que mantiene sincronizados a todos los dispositivos del bus. De esta forma, se elimina la necesidad de que cada dispositivo tenga su propio reloj, de tener que acordar una velocidad de transmisión y mecanismos para mantener la transmisión sincronizada (como en UART)

Para poder realizar la comunicación con solo un cable de datos, el bus I2C emplea una trama (el formato de los datos enviados) amplia. La comunicación costa de:

- 7 bits a la dirección del dispositivo esclavo con el que queremos comunicar.
- Un bit restante indica si queremos enviar o recibir información.
- Un bit de validación.
- Uno o más bytes son los datos enviados o recibidos del esclavo.
- Un bit de validación.

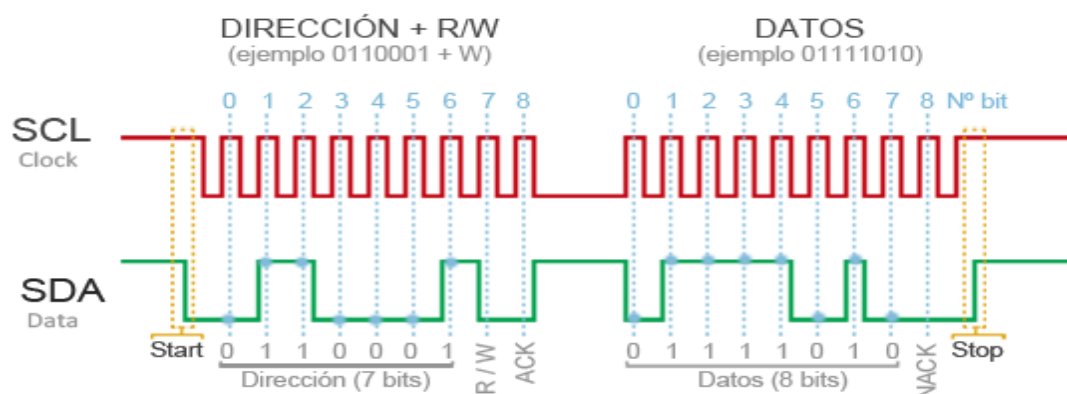


Figura 2-21: Trama de comunicación i2c

Fuente: Llamas, 2016

Con estos 7 bits de dirección es posible acceder a 112 dispositivos en un mismo bus. Por lo que restan solo 8 bits de datos, en general la velocidad del bus I2C es reducida. La velocidad estándar de transmisión es de 100Mhz. (Llamas, 2016)

2.12. Sensor ultrasónico HC-SR04

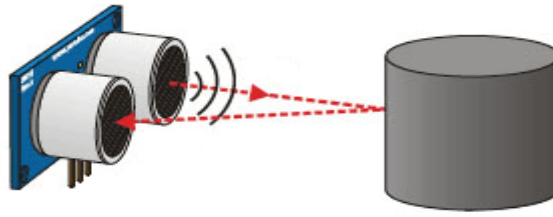
Los sensores de ultrasonido son muy frecuentes en los robots móviles (particularmente) y de forma significativa en los AUVs, por sus buenas propiedades de medición en entornos acuáticos y sirven para detectar objetos y medir distancias. Se utilizan para construir mapas del entorno y evitar obstáculos.

El sensor ultrasónico es un dispositivo de medición de distancia que se basa en las propiedades magnetostrictivas de determinados materiales. Una lámina de material magnetostrictivo o membrana tiene la propiedad de deformarse mecánicamente y generar ultrasonidos al ser excitada por una corriente eléctrica. El efecto contrario también se produce, es decir, que una vibración mecánica produce una corriente eléctrica. Por tanto, estos sensores emiten una radiación ultrasónica que rebota en los obstáculos del entorno y captan los ecos recibidos. (Llamas, Luis, 2016)

El HC-SR04 es un sensor capaz de detectar objetos que se encuentren en un rango de 2 a 450 cm, se destaca por su bajo consumo, gran precisión y bajo precio cuenta con las siguientes características:

- Dimensiones del circuito: 43 x 20 x 17 mm
- Tensión de alimentación: 5 Vcc
- Frecuencia de trabajo: 40 KHz
- Rango máximo: 4.5 m
- Rango mínimo: 1.7 cm
- Duración mínima del pulso de disparo (nivel TTL): 10 μ S.
- Duración del pulso eco de salida (nivel TTL): 100-25000 μ S.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 mS.

Para calcular la distancia se sabe que la velocidad del sonido es 343 m/s en condiciones de temperatura 20 °C, 50% de humedad, presión atmosférica a nivel del mar, es decir, el sonido tarda 29,2 microsegundos en recorrer un centímetro. Por tanto, podemos obtener la distancia a partir del tiempo entre la emisión y recepción del pulso mediante la siguiente ecuación.



$$\text{Tiempo} = 2 * (\text{Distancia} / \text{Velocidad})$$
$$\text{Distancia} = \text{Tiempo} \cdot \text{Velocidad} / 2$$

Figura 2-22: Formula para distancia de Sensor Ultrasónico

Fuente: Llamas, Luis, 2015

CAPÍTULO III

3. DISEÑO E IMPLEMENTACIÓN

El diseño del robot se realizó basándose en la investigación de los diferentes tipos de métodos para inspeccionar las líneas eléctricas, pero principalmente partiendo de los proyectos de robot con propósitos a fines, tanto de nivel experimental, así como de nivel comercial.

Por lo que se puntualizó las siguientes características que el robot debe tener para cumplir con sus propósitos:

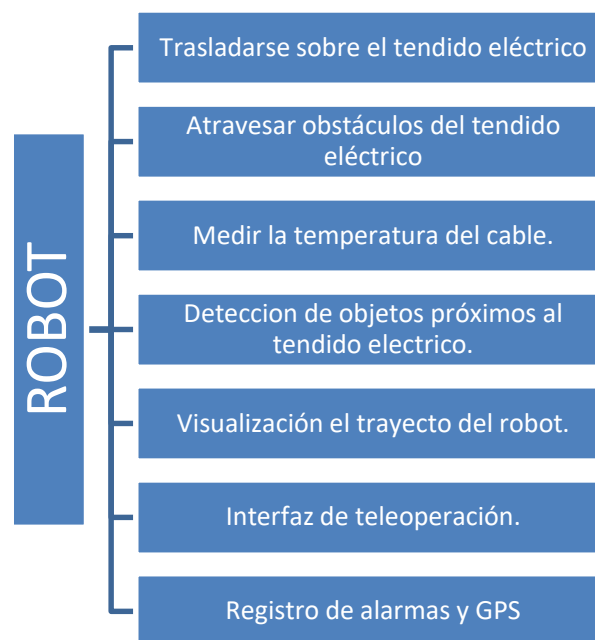


Figura 3-1: Características de diseño del Robot.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Para el diseño preliminar del sistema, se ha dividido en las siguientes secciones:

- a) Diseño estructura
- b) Diseño electrónico
- c) Diseño interfaz

3.1. Diseño estructura

El diseño del modelo de la estructura inicial consta de dos brazos grises los cuales, son los que sujetan el robot al cableado eléctrico, dos brazos dorados los cuales sujetan el robot cuando se necesita pasar un obstáculo, los cuatro brazos se encuentran sujetos a la misma base, y finalmente consta de dos ruedas que son las que están en contacto con el cableado eléctrico y brindan movimiento al robot.

Las dimensiones, de los brazos centrales, permiten pasar obstáculos de hasta 28cm, además se necesita una medida mínima de 75cm de cableado, por la longitud del robot.

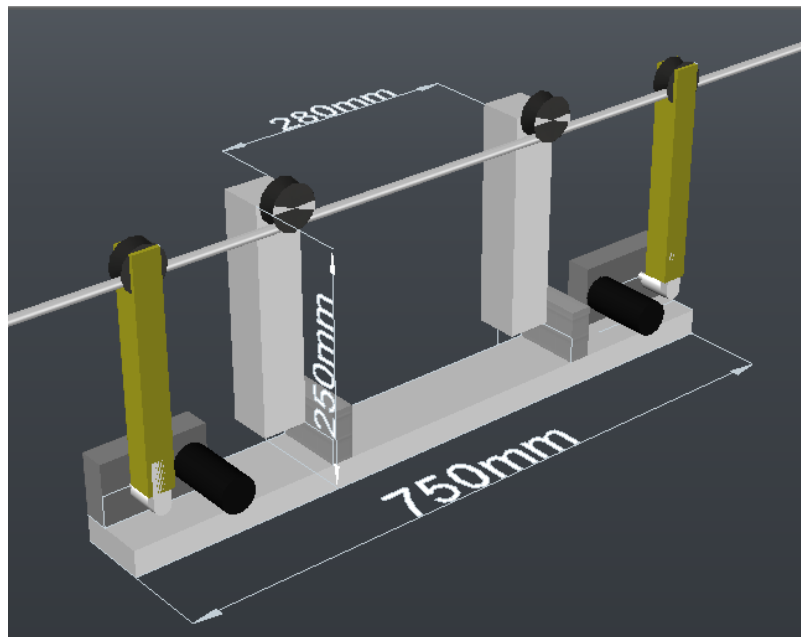


Figura 3-2: Principales medidas del robot

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.1.1. Cálculo torque de brazo dorado 1 y 2:

$$\vec{\tau} = \vec{r} \times \vec{F}$$

Donde: $\vec{\tau}$ es el vector torque.

\vec{r} es la distancia perpendicular desde el eje al extremo del brazo.

\vec{F} es el vector fuerza.

Tomando en cuenta que todas las variables son vectores, se tiene un producto vectorial; por lo tanto, el módulo del torque es el siguiente:

$$\tau = r \cdot F \sin \theta$$

La fuerza que tiene que vencer el motor es el peso del robot, por lo tanto:

$$\tau = r \cdot m \cdot g \cdot \sin \theta$$

Donde: τ es el torque.

r es la longitud del robot.

m es la masa del robot.

g es la gravedad $9,8 \frac{m}{s^2}$

θ es el menor ángulo formado entre r y F .

Reemplazando los valores, tenemos:

$$\tau = 0,7m \cdot 1,5Kg \cdot 9,8 \frac{m}{s^2} \cdot \sin 90$$

$$\tau = 10.29 [N.m]$$

Por las pérdidas existentes que pueden existir se dimensiona el 25% más del torque calculado.

$$\tau = 12.86 [N.m]$$



Voltaje	12 VDC
Amperaje	3A max
Velocidad	6 RPM
Torque	25lbf.ft / 35Nm

Figura 3-3: Motor para brazo dorado 1 y 2

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.1.2. Cálculo torque de brazo gris 1 y 2:

Partiendo de la ecuación para calcular el módulo del torque.

$$\tau = r \cdot F \sin \theta$$

La fuerza que tiene que vencer el motor es el peso del brazo incluyendo el peso del motor de las ruedas, por lo tanto:

$$\tau = r \cdot m \cdot g \cdot \sin \theta$$

Donde: τ es el torque

r es la longitud del brazo gris

m es la masa del robot

g es la gravedad $9,8 \text{ m/s}^2$

θ es el menor ángulo formado entre r y F

Reemplazando los valores, tenemos:

$$\tau = 0,25m \cdot 0,25Kg \cdot 9,8 \text{ m/s}^2 \cdot \sin 90$$

$$\tau = 0,61 \text{ [N.m]}$$

Por las pérdidas existentes que pueden existir se dimensiona el 25% más del torque calculado.

$$\tau = 0.76 \text{ [N.m]}$$



Voltaje	12 VDC
Velocidad	9 RPM
Reduccion	1/950

Figura 3-4: Motor para brazo gris 1 y 2

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

El motor comercial seleccionado tiene una caja reductora que entrega un torque de 8,5Kg.cm (0.8335 N.m)

3.1.3. Cálculo de la dimensión y motor de las ruedas.

Los brazos internos del robot tienen en su extremo dos ruedas tipo polea, que soportan un peso de hasta 50Kg que permiten trabajar con diámetros de hasta 23mm, aceptando un conductor eléctrico de hasta calibre 800.

MCM o kcmil	Circular mils	Área		Diametro	
		mm ²	in ²	mm	in
250	250000	126.677	0.1963	12.70	0.500
300	300000	152.012	0.2356	13.91	0.548
350	350000	177.348	0.2749	15.03	0.592
400	400000	202.683	0.3142	16.06	0.632
450	450000	228.018	0.3534	17.04	0.671
500	500000	253.354	0.3927	17.96	0.707
600	600000	304.025	0.4712	19.67	0.775
700	700000	354.695	0.5498	21.25	0.837
750	750000	380.031	0.5890	22.00	0.866
800	800000	405.366	0.6283	22.72	0.894
900	900000	456.037	0.7069	24.10	0.949
1000	1000000	506.708	0.7854	25.40	1.000

Figura 3-5: Medidas de conductores eléctricos de media y alta tensión.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

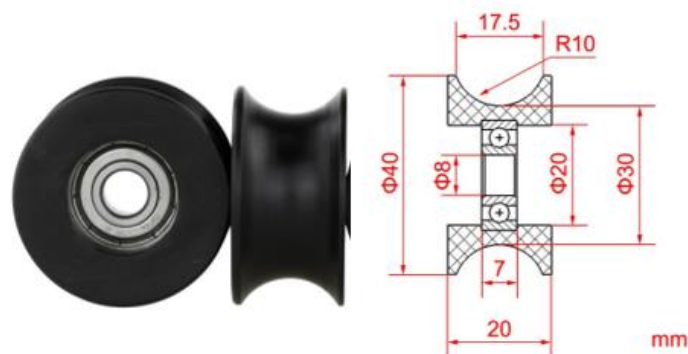


Figura 3-6: Dimensiones de rueda tipo polea con ranura.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Los motores de las ruedas del robot tienen caja reductora y tienen 180RPM, y un consumo de 250mA, el cual es uno de los componentes que requieren más energía, tomando en

cuenta que van a estar en continuo funcionamiento a diferencia de los motores de los brazos que funcionan solo por períodos



Voltaje	12 VDC
Velocidad	180 RPM
Corriente	250mA

Figura 3-7: Motor para ruedas del robot

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Los motores entregan la siguiente velocidad de desplazamiento del robot, en función de las medidas de las ruedas.

$$V = r \cdot \omega$$

Dónde: V es la velocidad lineal.

r es el radio de la rueda

ω es la velocidad angular del motor

Reemplazando los valores se tiene los siguientes resultados:

$$V = 0,015m \cdot 180RPM$$

$$V = 0,015m \cdot 18,84 rad/s$$

$$V = 0,015m \cdot 1130,4 rad/min$$

$$V = 16,96 m/min$$

3.1.4. Modelado 3D y simulación virtual

El modelado y simulación se realizó con los Software AutoCAD y V-REP respectivamente.

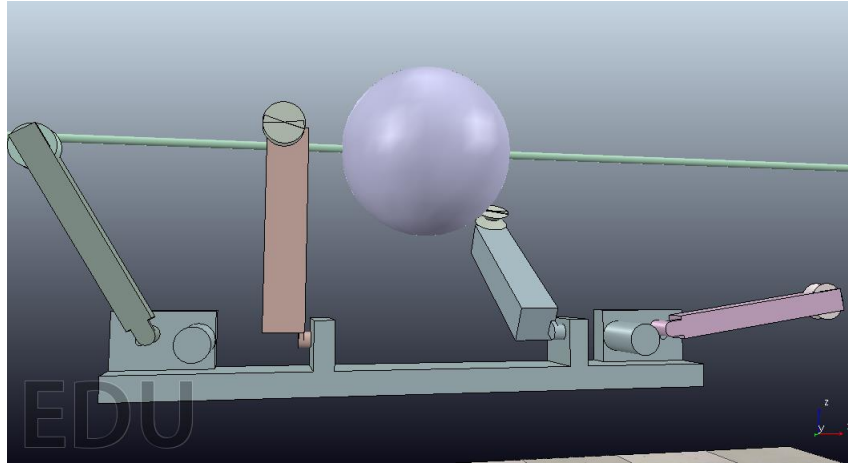


Figura 3-8: Simulación RILE (Robot Inspector De Líneas Eléctricas)

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Para la modelación y simulación de los movimientos de brazos y articulaciones del robot, se ha ocupado el software: Virtual Robot Experimentation Platform, V-REP PRO edu, el cual permite diseñar paso a paso robots móviles y no móviles, pudiendo ser manipulados automatizados o controlados mediante un Script de Programación, también por medio de una interface del propio software o de terceros y permitiendo además el control del robot real, entre muchas de sus características más importantes y relevantes.

3.1.4.1. *Modelado e importación de robot en 3d.*

El software V-REP permite el diseño 3D de formas básicas como esferas, cilindros, discos, etc. Denominadas formas primitivas. Por lo que se utilizó el software AUTOCAD 3D para el modelado del robot.

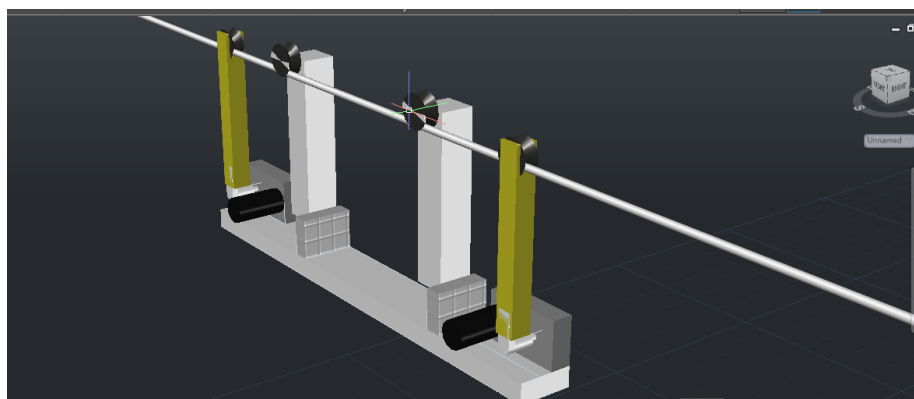


Figura 3-9: Diseño en 3D de la estructura del robot

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Una vez realizado el diseño de las partes que componen el robot, se utiliza el software AUTOCAD 3D para exportar los objetos en 3D en formato .stl, para posteriormente importarlos en V-REP, en donde se deben cargar cada parte del robot de manera individual.

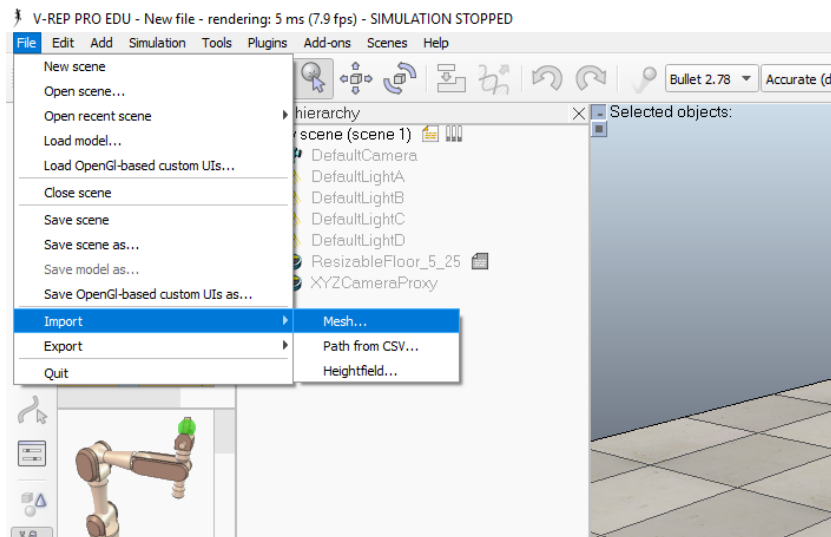


Figura 3-10 Exportar diseño 3D a Vrep

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

A continuación, y si es necesario se puede escoger las unidades con las que se desea trabajar y la orientación, en función de cómo se realizó el diseño en AUTOCAD 3D, esto también puede ser realizado después de cargar todos los elementos. Ya que el diseño en Autocad fue realizado con las medidas reales, no se necesitó ningún tipo de escalamiento.

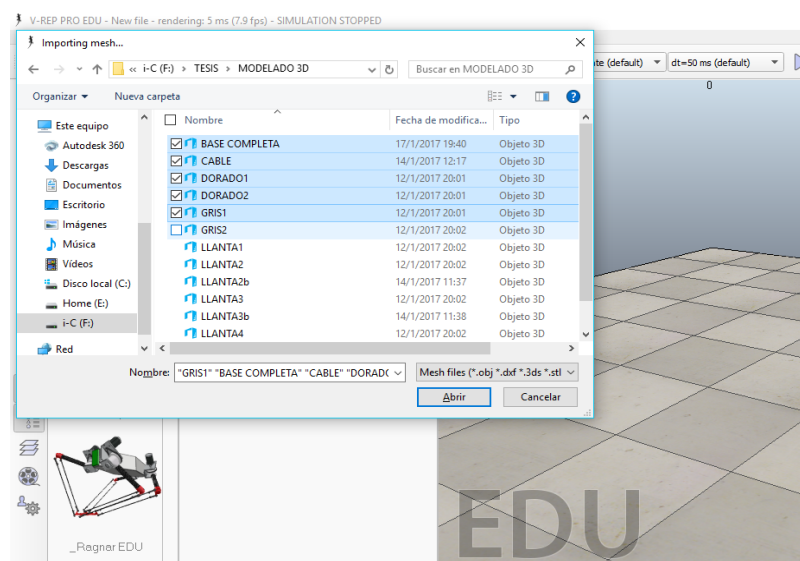


Figura 3-11: Diseño 3D de robot por partes.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Si todas las partes modeladas en AUTOCAD 3D, fueron dibujadas en las posiciones correspondientes para formar el robot, no existirá la necesidad de rotar o mover las partes, sino que más bien se importaran en su posición para ensamblar el robot. Caso contrario se deberá conocer la posición en el espacio de cada una de las partes que conforman el robot para ser configuradas mediante las herramientas de V-REP.

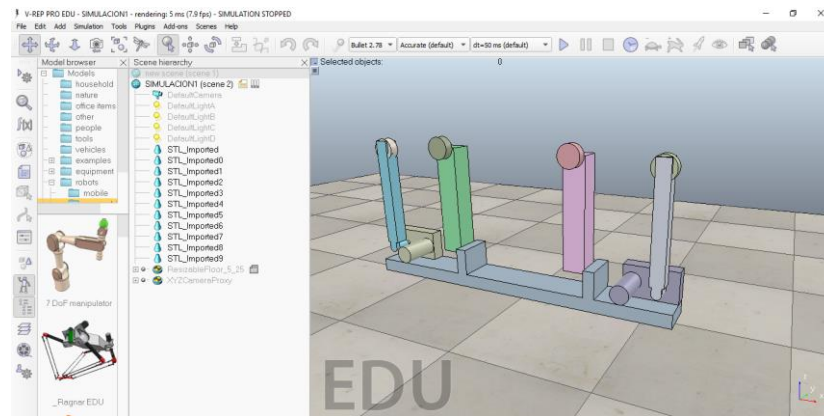


Figura 3-12: Diseño 3D exportado en Vrep

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Una vez que se realizó de manera correcta la importaciones de las partes modeladas en formato .stl, se podrá ver el robot en V-REP, tal se diseñó en AUTOCAD 3D, los nombres por defecto serán del tipo STL_Imported, los mismo que podrán ser modificados para poder identificarlos de una mejor manera.

3.1.4.2. Creación De Articulaciones

V-REP permite trabajar con tres tipos de juntas, de revolución, prismática o esférica. En el caso del robot RILE, todas las articulaciones son del tipo revolución. Para la creación de articulaciones nos dirigimos al menú Add, Joint, y seleccionamos Revolute. El robot RILE consta de 6 articulaciones del tipo revolución, las mismas que se encuentran ubicadas en los ejes de cada uno de los motores.

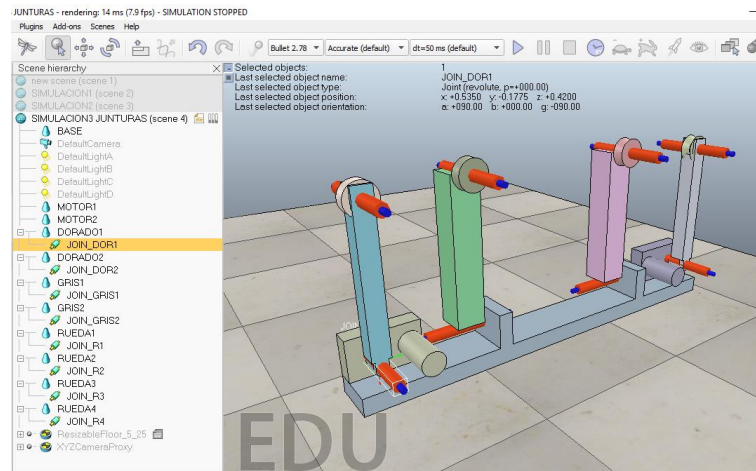


Figura 3-13: Creación de articulaciones Vrep

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

En la Figura 3-13 se puede observar ocho articulaciones, de las cuales seis serán activadas mediante motores, y las otras dos son solo ejes sin motores. Las medidas de cada una de las articulaciones pueden ser modificadas en longitud y diámetro en la ventana de Propiedades de cada objeto, en la sección de Visual Properties.

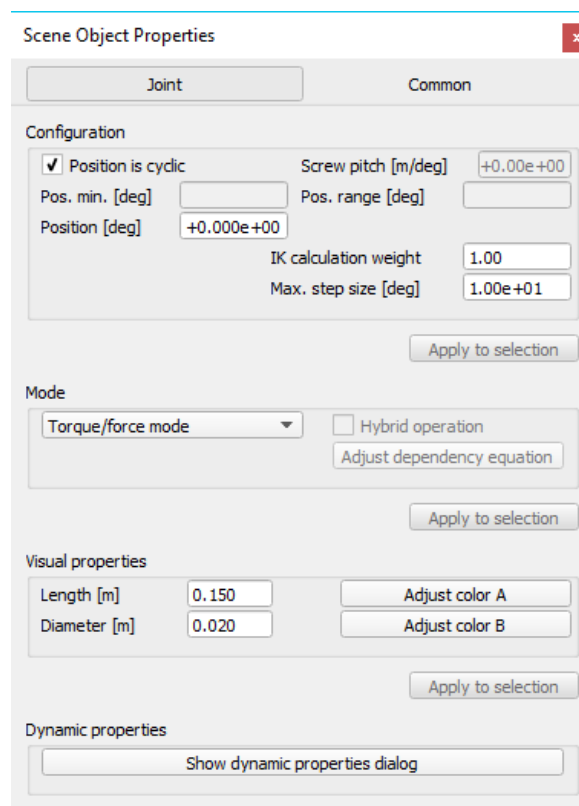


Figura 3-14: Propiedades de los objetos del diseño

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

En la misma ventana en la pestaña de Common, de ser necesario se pueden ocultar estas articulaciones en la sección de Visibility. Para el caso del robot RILE sus articulaciones no son cíclicas por lo que se deberá deseleccionar esta opción que viene por defecto.

3.1.4.3. Creación de la relación articulación-objeto

Como se había mencionado V-REP permite la creación de objetos denominados primitivos, los cuales tiene una geometría pura, pero los objetos importados tienen una geometría no pura, lo cual deberá tomarse en cuenta para crear la relación entre las articulaciones y los objetos importados.

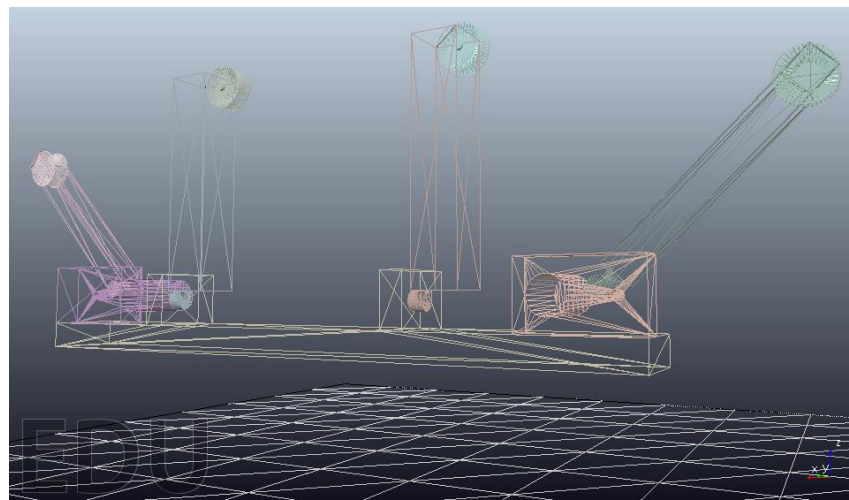


Figura 3-15: Objetos primitivos de geometría pura.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Los objetos estáticos no pueden ser directamente actuados a través de articulaciones controladas por par/fuerza, si no que su posición y orientación dependerá de la posición y orientación de otros objetos. (Armesto, 2016). De lo anteriormente citado, se concluye que para poder realizar movimientos de las partes del robot que fueron importadas, se deberán crear objetos puros o que permitan ser configurados como dinámicos y simular un motor en las articulaciones para su movimiento.

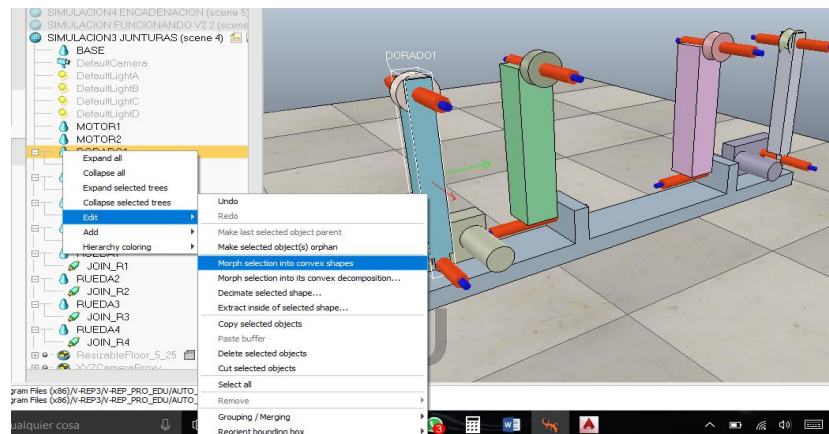


Figura 3-16: Creación de Objetos dinámicos y asignación de articulaciones

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

La creación de objetos dinámicos, se realiza a partir de los objetos importados, para lo cual se duplicarán estos objetos, pudiendo realizarlo copiando y pegando cada uno de los objetos, o realizando nuevamente la importación de cada uno de ellos.

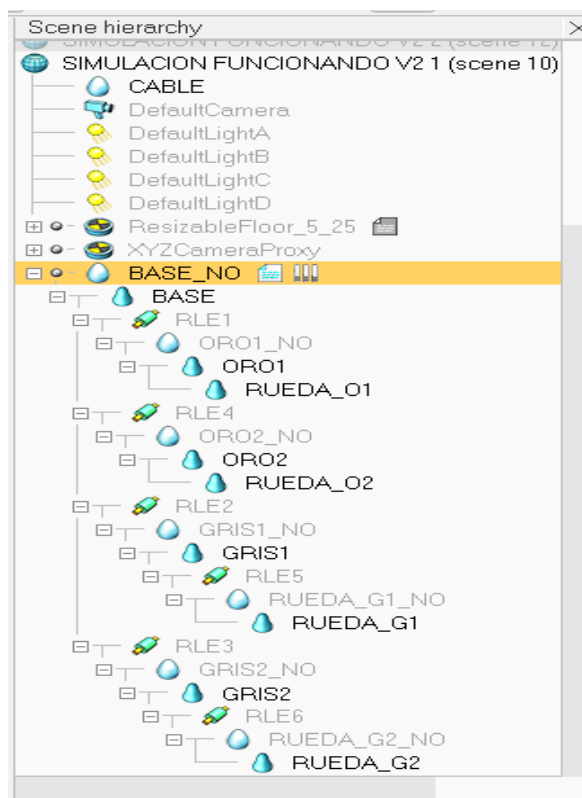


Figura 3-17: Asignación de jerarquías en articulaciones y objetos

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Una vez que tenemos cada uno de los objetos duplicados, a uno de ellos se lo convertirá en objeto dinámico, haciendo clic derecho en el nombre en Edit, Morph selection into convex shapes.

Visualmente el objeto convertido no tendrá la misma estética ni forma del modelado realizado en AUTOCAD 3D, por lo que procederemos a ocultar el objeto, ya que solo se utilizará como enlace entre la junta y el objeto no dinámico. Por lo que la relación quedaría de la forma: ARTICULACIÓN -> OBJETO DINÁMICO -> OBJETO ESTÁTICO. Cada una de las articulaciones se encuentran vinculadas a la base del robot, como se muestra a continuación:

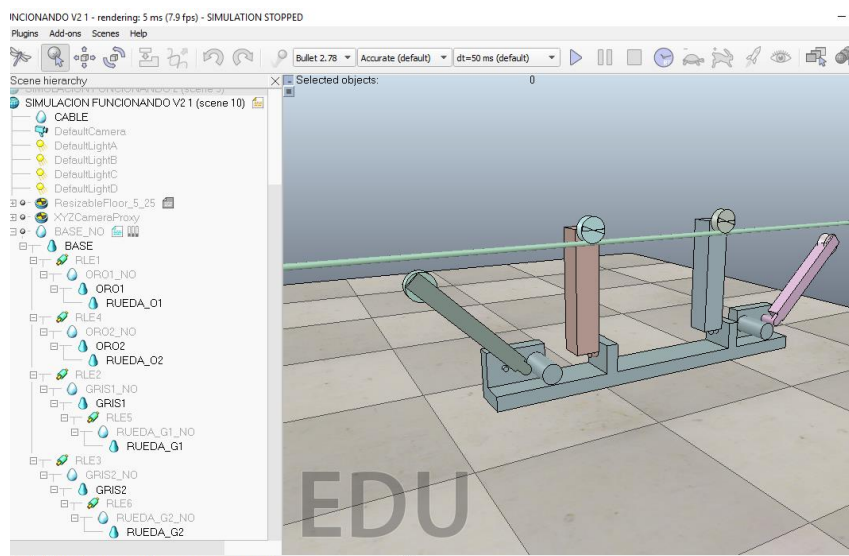


Figura 3-18: Diseño final del robot configurado para simulación en Vrep

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Las configuraciones de las articulaciones deberán estar configuradas en modo Torque/Fuerza, ya que su movimiento se realiza mediante motores.

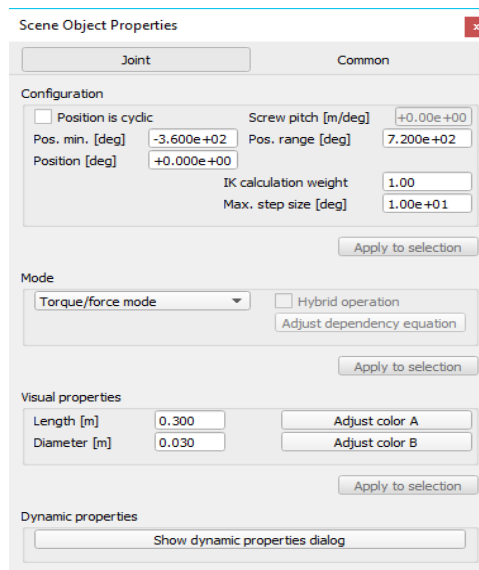


Figura 3-19: Configuración de articulaciones en modo Torque

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

En la sección de propiedades dinámicas de las articulaciones, se habilitará el motor y se configurará su máximo Torque.

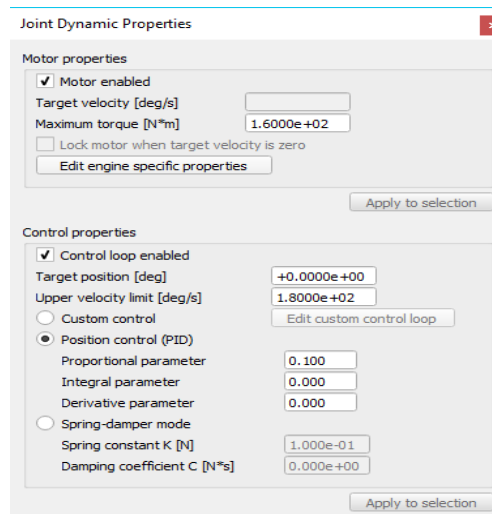


Figura 3-20: Habilidad de motor y torque

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

En la configuración de los objetos dinámicos, en la sección de Propiedades Dinámicas, se activará la sección de Body is Dynamic y se modificará los momentos de inercia que vienen por defecto, además de la masa del objeto.

En esta misma sección se habilitará la opción de Body is responsable, esta sección habilita la característica del objeto para poder responder a otros objetos o relacionarse con otros objetos.

Rigid Body Dynamic Properties

☒ Body is responsible

Local responsible mask ☐ ☐ ☐ ☒ ☒ ☒ ☒

Global responsible mask ☒ ☒ ☒ ☒ ☒ ☒ ☒

Material

☒ Body is dynamic

☐ Start in sleep mode ☐ Set to dynamic if gets parent

Mass

Mass [kg]

Principal moments of inertia / mass

X [m^2]

Y [m^2]

Z [m^2]

Pos./orient. of inertia frame & COM relative to shape frame

X [m]	<input type="text" value="+4.768e-07"/>	Alpha [deg]	<input type="text" value="-1.51e+01"/>
Y [m]	<input type="text" value="-3.278e-07"/>	Beta [deg]	<input type="text" value="+5.77e+01"/>
Z [m]	<input type="text" value="+3.353e-08"/>	Gamma [deg]	<input type="text" value="-1.31e+01"/>

Figura 3-21: Propiedades dinámicas del Cuerpo

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Finalmente, la configuración de el o los objetos estáticos que son los objetos visibles que son aquellos que se modelaron en AUTOCAD 3D, tendrán desactivadas las opciones de la sección de propiedades dinámicas. Ya que estos objetos dependerán del movimiento de los objetos dinámicos a los cuales están relacionados.

☐ **Body is responsible**

Local responsible mask

☒
☒
☒
☒
☒
☒
☒
☒

Global responsible mask

☒
☒
☒
☒
☒
☒
☒
☒

Material

☐ **Body is dynamic**

☐ Start in sleep mode
☐ Set to dynamic if gets parent

Mass

Principal moments of inertia / mass

Pos./orient. of inertia frame & COM relative to shape frame

X [m]	+0.000e+00	Alpha [deg]	+0.00e+00
Y [m]	+0.000e+00	Beta [deg]	+0.00e+00
Z [m]	+0.000e+00	Gamma [deg]	+0.00e+00

Figura 3-22: Configuración de visibilidad de objetos

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.1.4.4. Programación de script de movimiento.

V-REP permite controlar los robots que se van a simular mediante un script propio de programación, que pueden estar escritos en los siguientes lenguajes de programación: C/C++, Python, Java, Lua, Matlab o Urbi. Con lo cual se consigue simular los movimientos de cada una de las articulaciones y brazos del robot.

Para la creación de un nuevo script, se accede al menú Scripts y a continuación Insert new Script, este script se debe asociar a uno de los objetos, en este caso se asoció a la BASE.

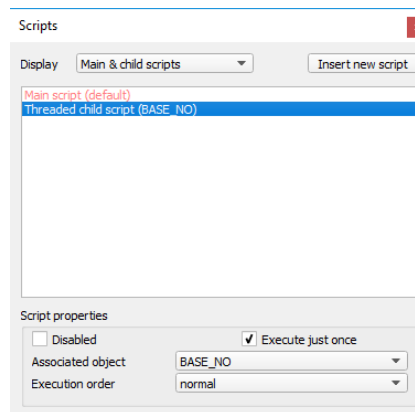


Figura 3-23: Creación de script de programación

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Para la programación de la secuencia de movimientos, las posiciones en x,y,z y sus velocidades y aceleraciones del robot y sus brazos, en el script se ocupan las siguientes funciones:

Tabla 3-1: Función API, simGetObjectHandle:

Description	Retrieves an object handle based on its name. The operation of this function depends on the current name suffix settings (see <code>simGetNameSuffix</code>, <code>simSetNameSuffix</code>, and the section on accessing general-type objects). See also <code>simIsValid</code> and <code>simGetObjectUniqueId</code>.
C synopsis	<code>simInt simGetObjectHandle(const simChar* objectName)</code>
C parameters	<code>objectName</code>: name of object
C return value	handle of object or -1 if operation was not successful
Lua synopsis	<code>number objectHandle=simGetObjectHandle(string objectName)</code>
Lua parameters	Same as C-function
Lua return values	Same as C-function

Fuente: (Coppeliarobotics, 2016)

Tabla 3-2: Función API, simRMLMoveToJointPositions

Description	<i>Moves (actuates) several joints at the same time using the Reflexxes Motion Library type IV. This function can only be called from child scripts running in a thread (since this is a blocking operation) and is not available from the C-API. See also simRMLMoveToPosition, and simRMLPos.</i>
Lua synopsis	<i>number result,table newPos,table newVel,table newAccel,number timeLeft=simRMLMoveToJointPositions(table jointHandles,number flags,table currentVel,table currentAccel,table maxVel,table maxAccel,table maxJerk,table targetPos,table targetVel,table direction)</i>
Lua parameters	<i>jointHandles: handles of the joints to actuate</i> <i>flags: RML flags. -1 for default flags.</i> <i>currentVel: the current velocity of the joints. Can be nil in which case a velocity vector of 0 is used.</i> <i>currentAccel: the current acceleration of the joints. Can be nil in which case an acceleration vector of 0 is used.</i> <i>maxVel: the maximum allowed velocity of the joints</i> <i>maxAccel: the maximum allowed acceleration of the joints</i> <i>maxJerk: the maximum allowed jerk of the joints</i> <i>targetPos: the desired target positions of the joints</i> <i>targetVel: the desired velocity of the joints at the target. Can be nil in which case a velocity vector of 0 is used.</i> <i>direction: the desired rotation direction for cyclic revolute joints: 0 for the shortest distance, -x for a movement towards negative values, +x for a movement towards positive values (n=(x-1) represents the number of additional turns). Can be nil or omitted, in which case a value of 0 is used for all joints.</i>
Lua return values	<i>result: 1 if the function call was successful</i> <i>newPos: the new positions of the joints</i> <i>newVel: the new velocities of the joints</i> <i>newAccel: the new accelerations of the joints</i> <i>timeLeft: the time left for additional calculations in current simulation time step</i>

Fuente: (Coppeliarobotics, 2016)

Las 2 funciones mencionadas son utilizadas para poder simular el movimiento de cada uno de los brazos.

Se crea un identificador de los objeto o brazos del robot RILE con la función simGetObjectHandle, para ello es necesario crear un vector que deberá tener el tamaño igual al número de brazos o articulaciones que se van a controlar. En el caso del Robot RILE son necesarias siete elementos, por lo que queda de la siguiente manera: BRAZOS={-1,-1,-1,-1,-1,-1,-1}, donde: BRAZOS es el nombre del vector de siete elementos vacíos, por lo cual se ocupa -1.

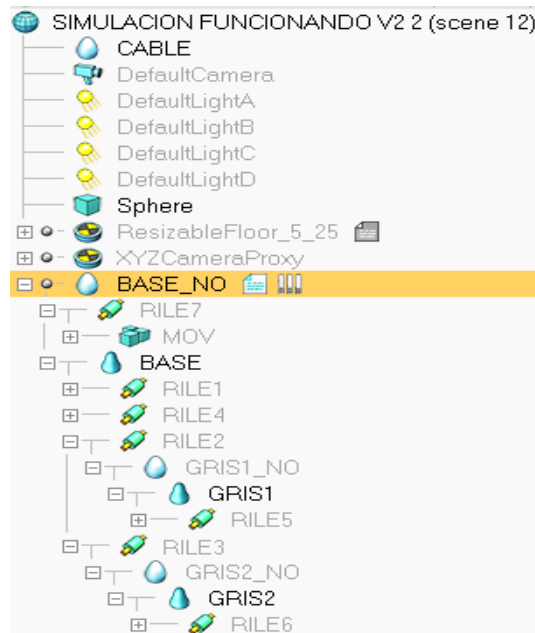


Figura: 3-24 Asignación de script al Robot

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

En el vector BRAZOS se asignará cada una de las siete articulaciones de nombre RILE1 hasta RILE7, por lo que se creará una función de secuencia repetitiva FOR para realizar las siete asignaciones en secuencia:

Para poder asignar las posiciones de cada brazo mediante la función `simRMLMoveToJointPositions`, es preciso crear todos los parámetros necesarios para el funcionamiento de la misma, así:

```

1  -----RILE ROBOT DE INSPECCION DE LINEAS ELECTRICAS-----
2
3  BRAZOS={-1,-1,-1,-1,-1,-1,-1}
4  for i=1,7,1 do
5      BRAZOS[i]=simGetObjectHandle('RILE'..i)
6  end
7
8  -- Configuración de los vectores iniciales del ROBOT RILE:
9  vel=20
10 acel=40
11 arrang=80
12 AVel={0,0,0,0,0,0,0}
13 AAcel={0,0,0,0,0,0,0}
14 maxVel={vel*math.pi/180,vel*math.pi/180,vel*math.pi/180,vel*math.pi/180,vel*math.pi/180,vel*math.pi/180,vel*math.pi/180}
15 maxAcel={accel*math.pi/180,accel*math.pi/180,accel*math.pi/180,accel*math.pi/180,accel*math.pi/180,accel*math.pi/180,accel*math.pi/180}
16 maxArrang={arrang*math.pi/180,arrang*math.pi/180,arrang*math.pi/180,arrang*math.pi/180,arrang*math.pi/180,arrang*math.pi/180,arrang*math.pi/180}
17 objVel={0,0,0,0,0,0,0}
18
19

```

Figura 3-25: Creación de Valores iniciales

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Estos valores serán comunes para todos los movimientos realizados por los brazos y articulaciones del robot RILE, contrario con lo que sucede con el vector posición, el cual

contiene la posición de cada una de las siete articulaciones, las cuales serán modificadas en función de la posición que se desee asignar a cada brazo.

```

19 -----PRIMERA ETAPA
20
21 Pos0={0,0,0,200*math.pi/180,200*math.pi/180,200*math.pi/180}
22
23 simRMLMoveToJointPositions (BRAZOS,-1,AVel,AAcel,maxVel,maxAcel,maxArrang,Pos0,objVel)
24
25 Pos1={0,0,0,14*math.pi/180,200*math.pi/180,200*math.pi/180}
26 simRMLMoveToJointPositions (BRAZOS,-1,AVel,AAcel,maxVel,maxAcel,maxArrang,Pos1,objVel)
27
28 Pos2={0,60*math.pi/180,0,14*math.pi/180,200*math.pi/180,200*math.pi/180}
29 simRMLMoveToJointPositions (BRAZOS,-1,AVel,AAcel,maxVel,maxAcel,maxArrang,Pos2,objVel)
30
31 Pos3={0,60*math.pi/180,0,14*math.pi/180,200*math.pi/180,200*math.pi/180}
32 simRMLMoveToJointPositions (BRAZOS,-1,AVel,AAcel,maxVel,maxAcel,maxArrang,Pos3,objVel)
33
34
35 Pos4={-60*math.pi/180,60*math.pi/180,0,14*math.pi/180,200*math.pi/180,200*math.pi/180}
36 simRMLMoveToJointPositions (BRAZOS,-1,AVel,AAcel,maxVel,maxAcel,maxArrang,Pos4,objVel)
37
38 Pos5={-60*math.pi/180,60*math.pi/180,0,14*math.pi/180,700*math.pi/180,700*math.pi/180}
39 simRMLMoveToJointPositions (BRAZOS,-1,AVel,AAcel,maxVel,maxAcel,maxArrang,Pos5,objVel)
40

```

Figura: 3-26 Secuencia de movimientos primera etapa

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Para realizar la simulación del movimiento del robot RILE se realizaron 14 vectores posición, es decir que se realizaran 14 movimientos.

```

40 -----REGRESAR 1RA ETAPA
41
42 Pos6={14*math.pi/180,60*math.pi/180,0,14*math.pi/180,700*math.pi/180,700*math.pi/180}
43 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos6,targetVel)
44
45
46 Pos7={14*math.pi/180,0,0,14*math.pi/180,700*math.pi/180,700*math.pi/180}
47 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos7,targetVel)
48
49 Pos8={14*math.pi/180,0,-60*math.pi/180,14*math.pi/180,700*math.pi/180,700*math.pi/180}
50 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos8,targetVel)
51
52 Pos9={14*math.pi/180,0,-60*math.pi/180,-60*math.pi/180,700*math.pi/180,700*math.pi/180}
53 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos9,targetVel)
54
55 Pos10={14*math.pi/180,0,-60*math.pi/180,-60*math.pi/180,1200*math.pi/180,1200*math.pi/180}
56 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos10,targetVel)
57
58 Pos11={14*math.pi/180,0,0*math.pi/180,-60*math.pi/180,1200*math.pi/180,1200*math.pi/180}
59 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos11,targetVel)
60
61 Pos12={14*math.pi/180,0,0*math.pi/180,0*math.pi/180,1200*math.pi/180,1200*math.pi/180}
62 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos12,targetVel)
63
64 Pos13={0,0,0,1200*math.pi/180,1200*math.pi/180,1200*math.pi/180}
65 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos13,targetVel)
66
67 Pos14={0,0,0,1500*math.pi/180,1500*math.pi/180,1500*math.pi/180}
68 simRMLMoveToJointPositions (BRAZOS,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Pos14,targetVel)

```

Figura: 3-27 Secuencia de movimientos 2da etapa

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

En la simulación final, además de las partes que componen el robot RILE se crearon dos objetos, un cilindro de gran extensión para simular el cableado de las líneas eléctricas, y una esfera para simular un obstáculo en las líneas eléctricas.

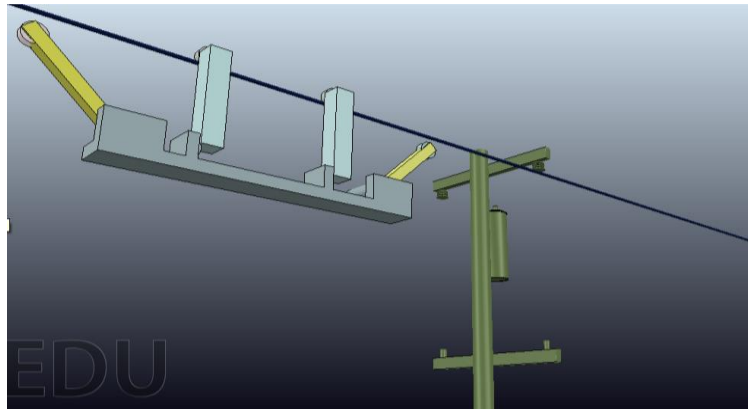
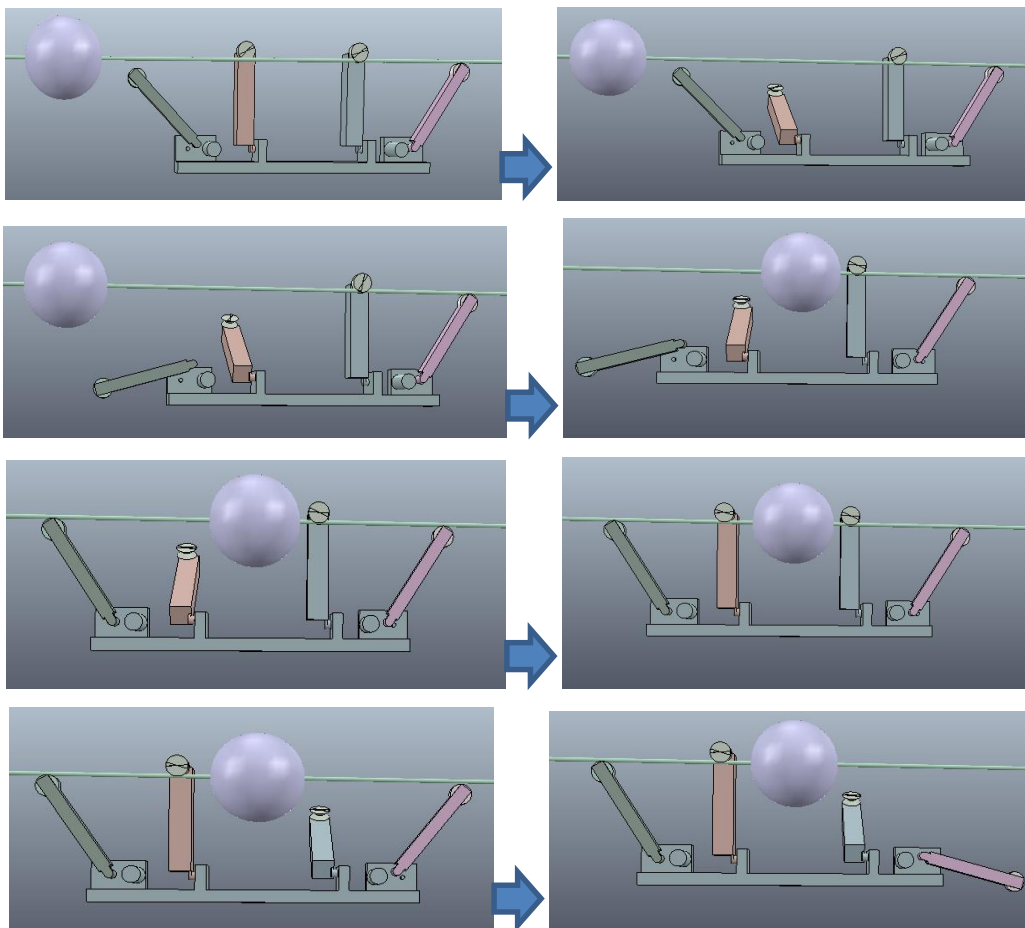


Figura: 3-28 Robot Rile 3D en simulación.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

La secuencia de movimientos para eludir un obstáculo en las líneas de transmisión, este compuesta como mínimo de 12 movimientos, que se muestran a continuación:



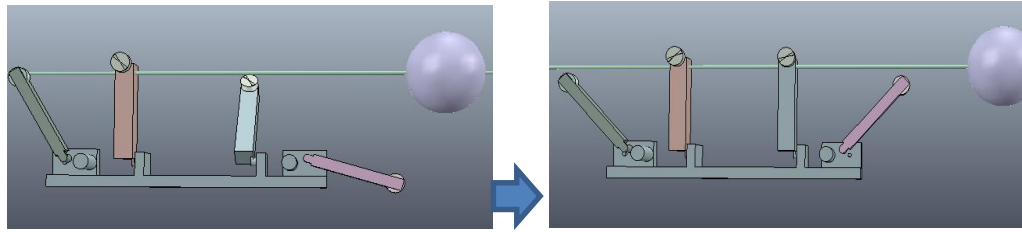


Figura 3-29: Secuencia de movimientos para eludir un obstáculo.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

En la secuencia de movimientos, se puede observar que todos los brazos cumplen una función específica para poder superar el obstáculo, así como también el activar las ruedas del robot, para que éste se desplace sobre el cableado eléctrico, mientras supera finalmente el obstáculo.

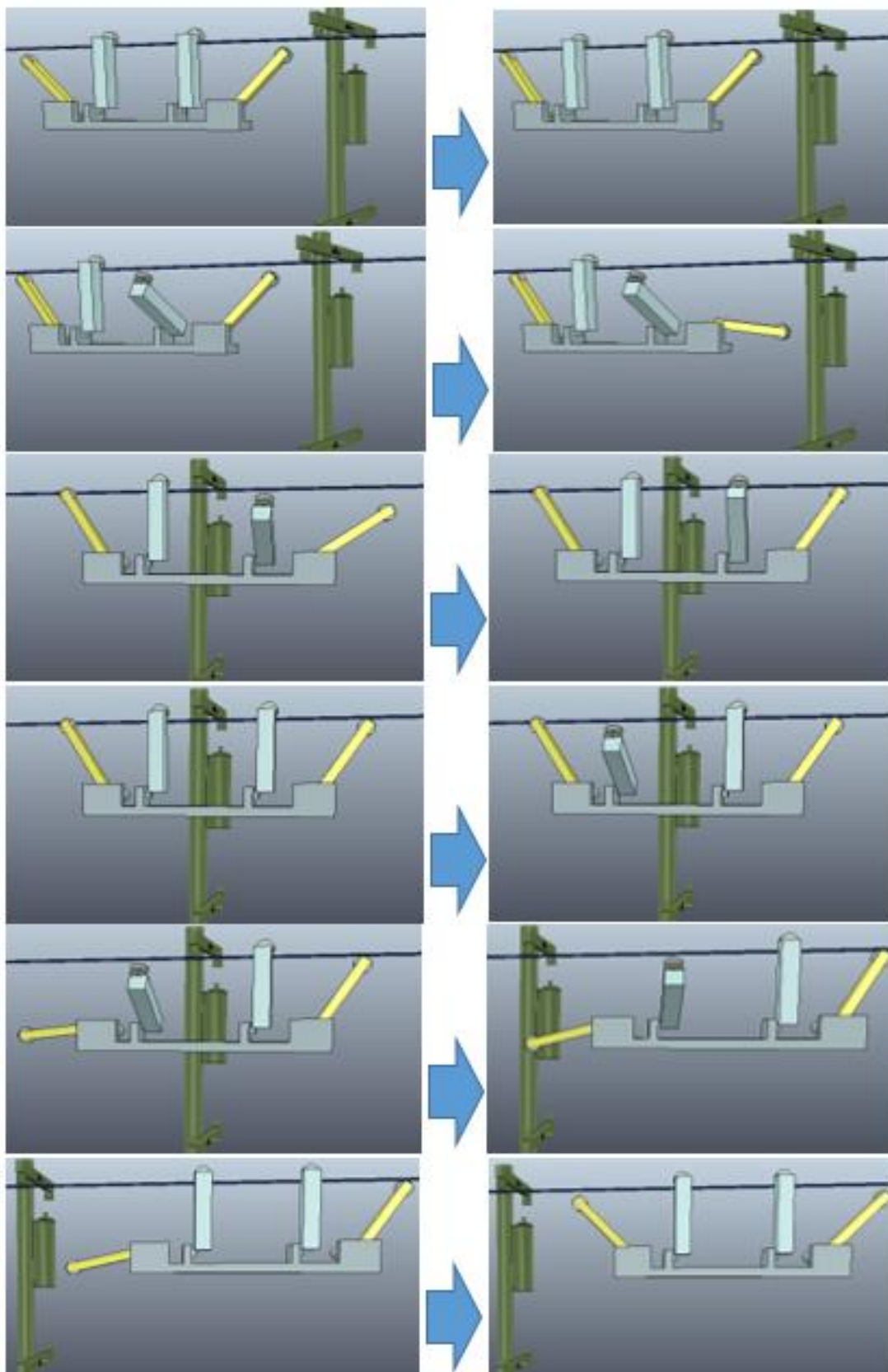


Figura: 3-30 Secuencia de movimientos para pasar un poste

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.2. Diseño electrónico

El robot móvil, contará con tres etapas básicas en un sistema:

Entradas: Sensores infrarrojos, Modulo de Posicionamiento Global, cámara ip.

Salidas: Etapa de Potencia, motoreductores.

Control: Microcontrolador Arduino Mega.

Además, el robot móvil contará con módulos de comunicación, descritos en la sección siguiente.

La alimentación de energía se realiza por medio de una batería capaz de alimentar todos sus componentes, dándole la autonomía necesaria al robot móvil.

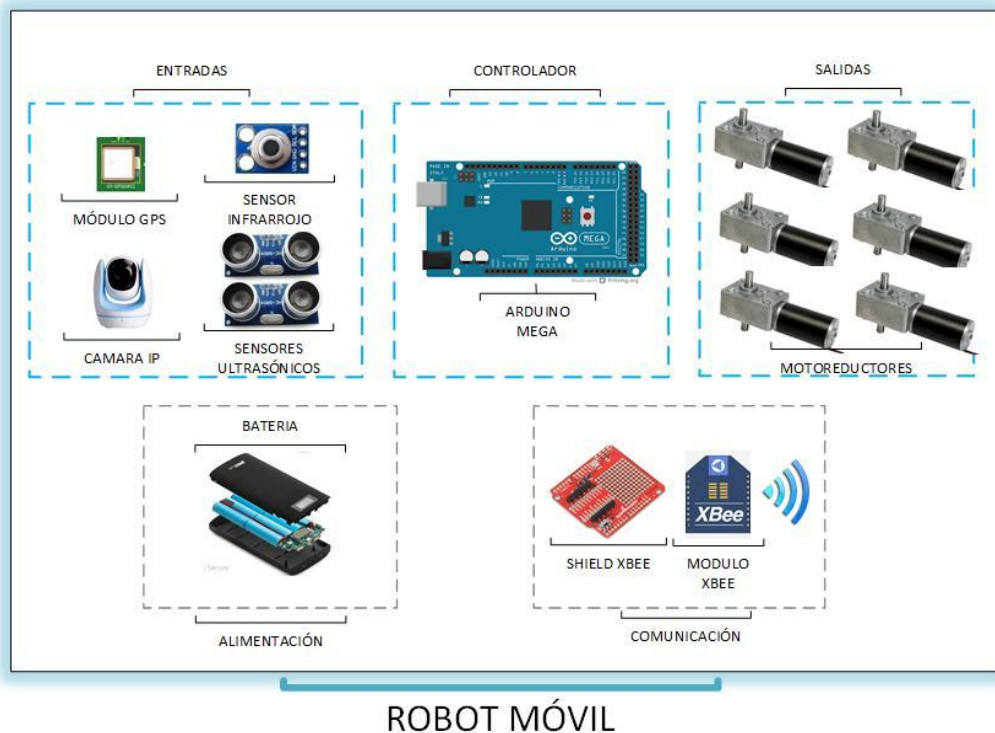


Figura 3-31: Componentes del robot móvil.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.2.1. Etapa de Control.

El hardware del controlador consiste en una placa Arduino Mega 2560, el cual tendrá en sus puertos digitales y analógicos los dispositivos de entrada y de salida, es decir los

sensores y actuadores del Robot, el cual se conectará a una placa de expansión (shield) para la comunicación XBee,

Los dispositivos conectados a la placa se muestran en la siguiente tabla:

Tabla 3-3: Tabla de conexión de dispositivos.

Dispositivo	Función	Puertos	Configuración IN/OUT
Driver 1: Motor1 Motor2	Brazo dorado 1 Brazo Gris 1	36-37 38-39	SALIDAS DIGITALES
Driver 2, Motor3 Motor4	Brazo dorado 2 Brazo Gris 2	34-35 32-33	SALIDAS DIGITALES
Driver 3, Motor 5 y 6	Ruedas	30-31	SALIDAS DIGITALES
Servomotor Control cámara	Giro 180 grados cámara IP	9	SALIDAS DIGITALES
Potenciómetro 1 para ángulo de giro	Brazo Gris 1	A10	ENTRADA ANALÓGICA
Potenciómetro 2 para ángulo de giro	Brazo Gris 2	A15	ENTRADA ANALÓGICA
Sensor Infrarrojo de Temperatura	Temperatura Ambiente y Temperatura del cable	SDA 20 SCL 21	COMUNICACIÓN i2c
Sensor Ultrasónico de Distancia 1	Distancia de objetos Lado Derecho	TRIG 26 ECHO 27	TRIG ECHO
Sensor Ultrasónico de Distancia 1	Distancia de objetos Lado Izquierdo	TRIG 52 ECHO 53	TRIG ECHO
Luz Indicadora LED RGB	Alarma de Temperatura, Sensor Derecho, Sensor Izquierdo	R 43 G 40 B 41	ENTRADA DIGITAL
Modulo GPS	Ubicación Latitud y Longitud	TX1 18 RX1 19	COMUNICACIÓN SERIAL 1

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.2.2. Diseño de la comunicación

Dado que el robot será operado de manera remota, la comunicación entre la interfaz de usuario y el robot móvil, tendrá comunicación de manera inalámbrica. Desde el robot se transmitirán dos tipos de información:

1. Transmisión de datos de sensores, actuadores y control.
2. Transmisión de video de cámaras ip.

La primera etapa de comunicacion se realizará mediante módulos de comunicación Xbee, y la segunda etapa mediante Wifi.

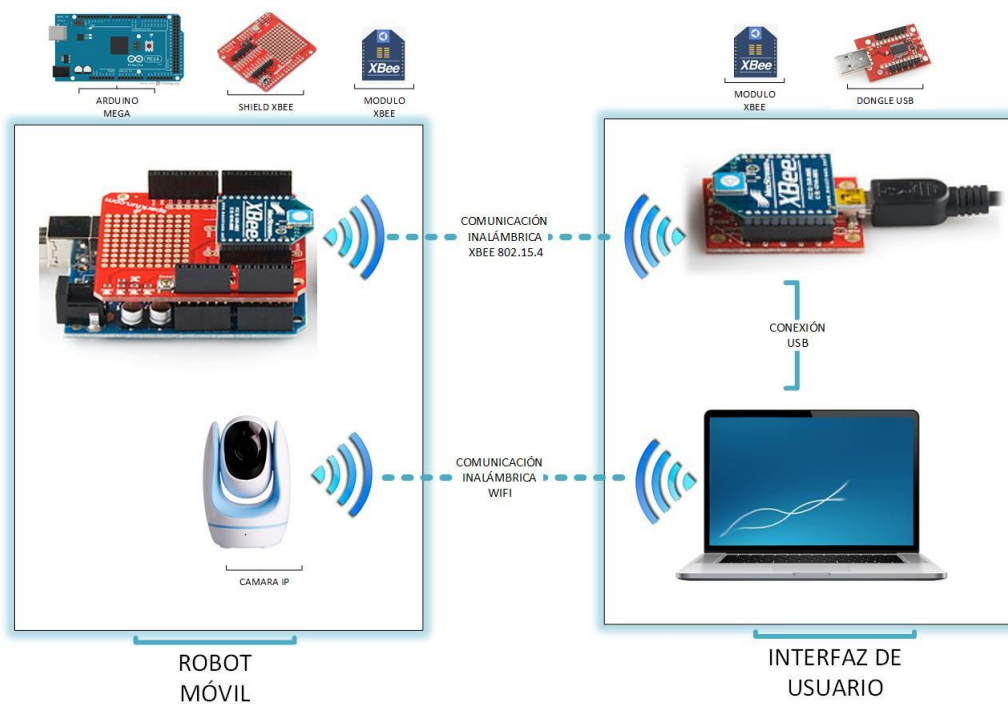


Figura 3-32: Diagrama de comunicación entre el robot y el operador.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2016

3.2.3. Configuración de los Módulos XBEE.

Los módulos de comunicación inalámbrica son dos Xbee Series 1, mediante el software X-CTU, se puede acceder a información y configuración mediante el PC, para lo cual es necesario un Xbee Explorer Usb,

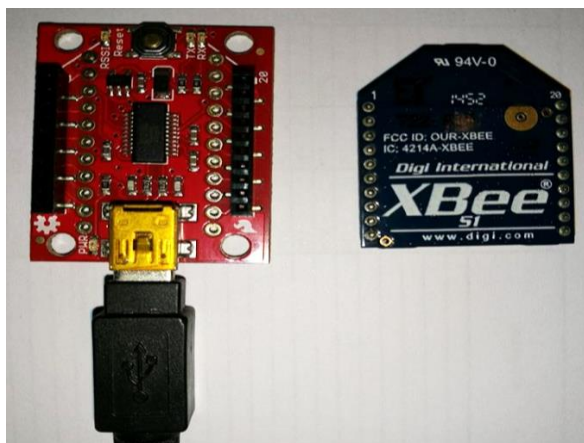


Figura 3-33: Módulo Xbee y Xbee Explorer USB

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2016

En el software se puede realizar un Test de conexión y comunicación con el dispositivo conectado, el cual aparecerá como USB Serial Port. Si la comunicación entre el PC y el dispositivo es correcta aparecerá la pantalla con la información correspondiente, además de su número de serie.

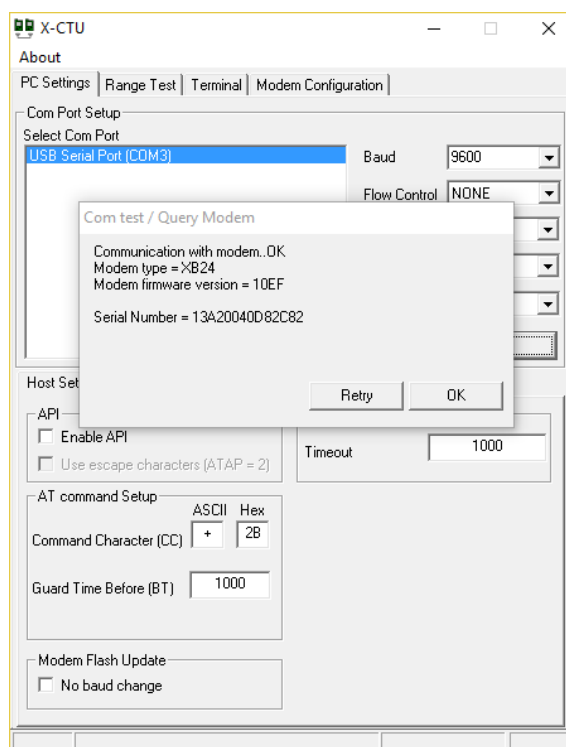


Figura 3-34: Software X-CTU para configurar módulos XBEE.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Para proceder a la configuración, se accede a la pestaña Modem Configuration, en la cual se modificarán los siguientes parámetros: DL, MY y CE.

En DL se ingresará la dirección del destino, en MY se colocará la dirección del dispositivo actual, además se deberá configurar en CE como end device. Finalmente, con el botón Write escribiremos las modificaciones en el dispositivo para que se guarden los cambios realizados.

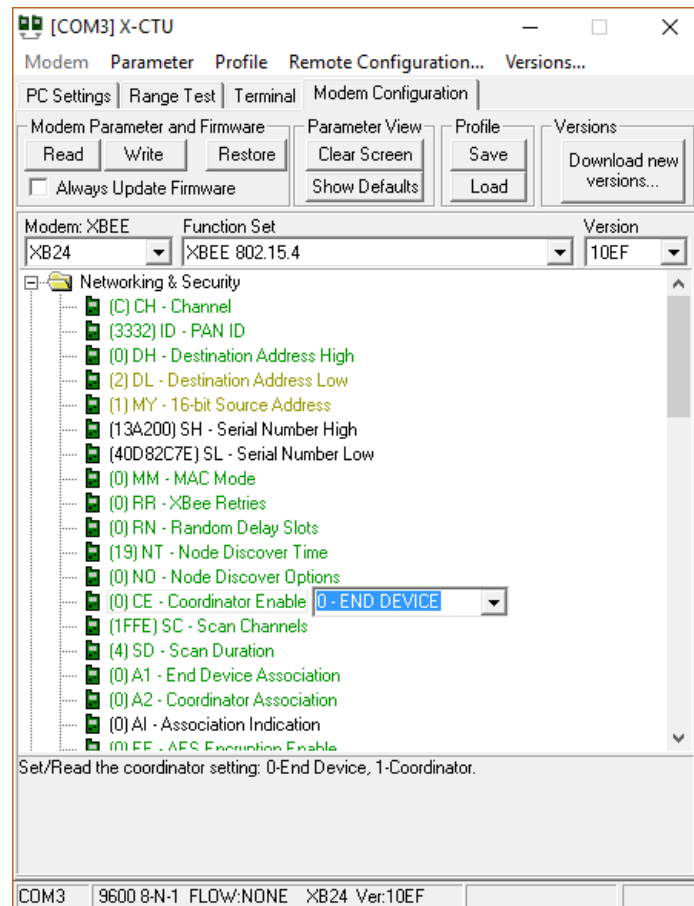


Figura 3-35: Configuración módulos XBEE.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.2.4. Configuración de cámara ip

La cámara ip instalada a bordo del robot, permite principalmente visualizar el recorrido del robot lo que permite la movilización y desplazamiento del mismo, además de poder ir visualizando físicamente el estado del cableado y sus accesorios.

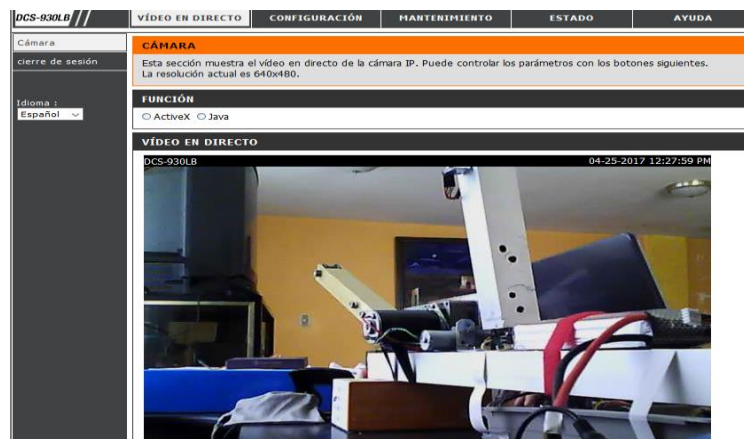


Figura 3-36: Transmisión en directo de cámara ip.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Es una cámara fija D-link modelo DCS930LB, con una resolución de un megapixel, a la misma que se configuró un ip estática para poder realizar una conexión directa al software y posteriormente a la interfaz de usuario.

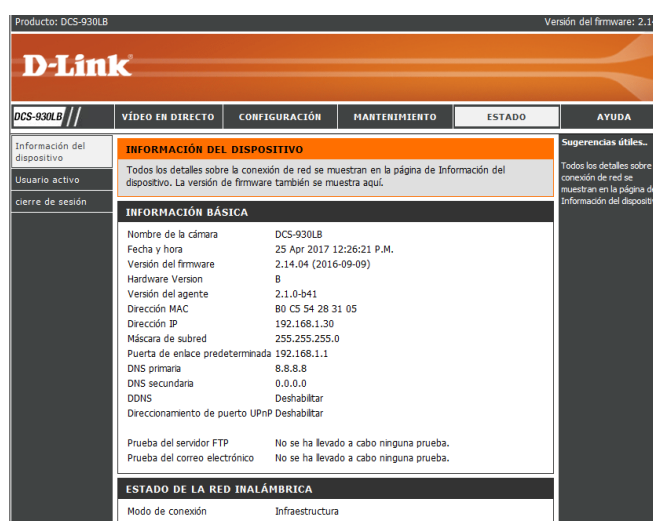


Figura 3-37: Configuración cámara ip

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.2.5. Etapa de Potencia.

Para el control de los brazos del robot y de las ruedas que darán movimiento al mismo, se necesita motores con la suficiente fuerza para poder realizar estos trabajos, por lo cual se ocupó motores con caja reductora.

Estos motores requieren una etapa para el control para el giro de los motores y también por la potencia que necesitan, la cual el Arduino Mega por sí solo no puede suministrar.

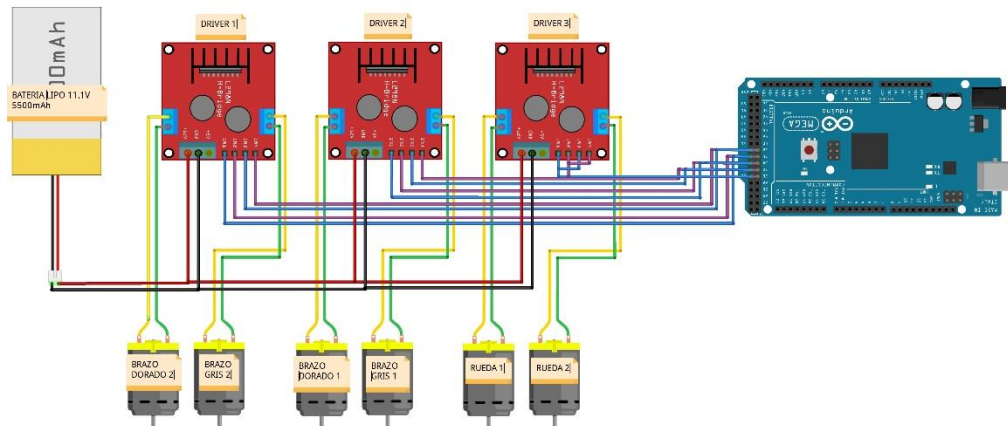


Figura 3-38: Diagrama de conexiones motores y drivers

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Por lo tanto, es necesario ocupar un driver que permite controlar dos motores de corriente continua, este módulo está basado en el chip L298N. El módulo tiene la salida A conformada por OUT1 y OUT2 y la salida B por OUT3 y OUT4. Los pines de habilitación son ENA y ENB respectivamente.

Las señales que controlan el movimiento de los motores son enviadas desde el controlador Arduino, y dependerán directamente de los botones de mando de la interfaz de usuario, siguiendo el siguiente diagrama de flujo general.

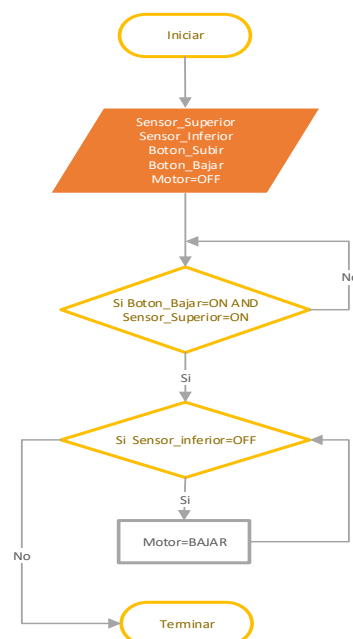


Figura 3-39: Diagrama de Flujo Control de Motores.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.2.6. Etapa de sensores y dispositivos de entrada.

3.2.6.1. Sensor de Temperatura

Para tomar la medida de la temperatura del cableado eléctrico, es necesario un sensor que no necesite estar en contacto con el cableado eléctrico, además de tener la habilidad de poder detectar cambios rápidos de temperatura.

Por lo que se eligió un sensor infrarrojo de temperatura, el cual se encuentra instalado en un brazo del robot a una distancia de 10cm de la rueda por donde va a recorrer el cableado. El mismo que entrega tanto la temperatura ambiente como la temperatura del objeto, es decir el cable eléctrico.

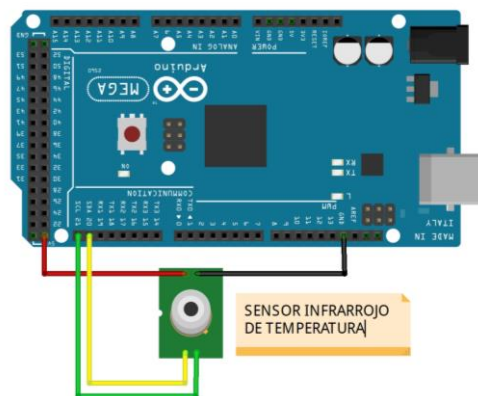


Figura 3-40: Diagrama de conexión sensor de temperatura.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

El sensor MLX90614 tiene dos métodos de comunicación: i2c y pwm; el Arduino Mega tiene un puerto i2c, en el cual se conectó el sensor. La comunicación i2c tiene mayor resolución frente al PWM,

La versión del sensor instalado es MLX90614ESF-AAA, el mismo que tiene las siguientes características principales:

Tabla 3-4 Tabla de características sensor IR

Tipo de sensor	Digital, Infrarrojo IR
Rango de temperatura	-70°C ~ 380°C (IR)
Voltaje de alimentación	4.5 V ~ 5.5 V
Precisión	± 0.5°C

Interfaz de comunicación	PWM, bus i2c
FOV	90°

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

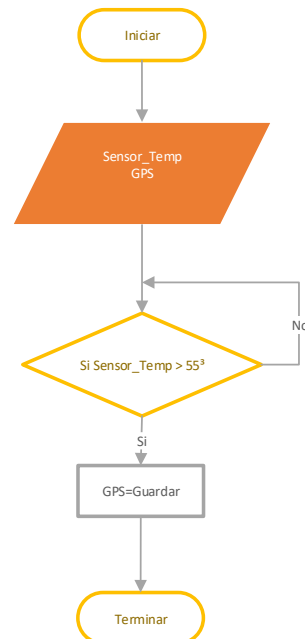


Figura 3-41: Diagrama de Flujo Sensor de Temperatura

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.2.6.2. Sensor de Distancia

Para la detección de posibles cuerpos que en futuro puedan ser un problema para el tendido eléctrico, se instalaron dos sensores de distancia, que además de indicar que existe un cuerpo, también indican a que distancia se encuentra el mismo.

Los sensores son ultrasónicos con el siguiente diagrama de conexión a la placa de control Arduino:

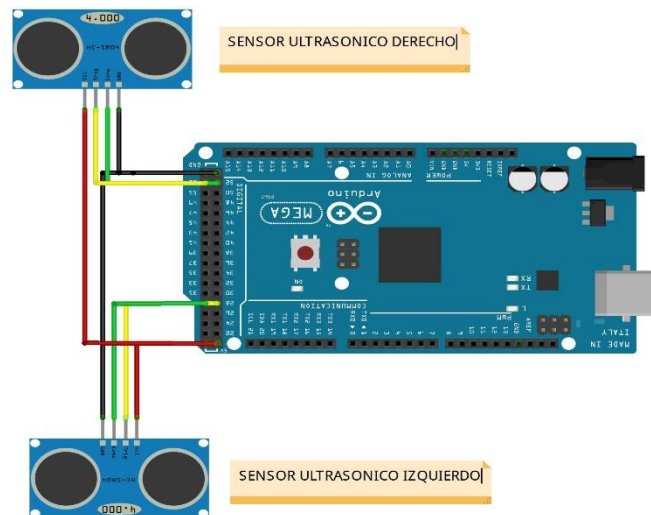


Figura 3-42: Diagrama de conexiones Sensores Ultrasónicos de distancia

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Básicamente funcionan emitiendo una señal ultrasónica y midiendo el tiempo en el que la señal viaja, choca contra un cuerpo y regresa nuevamente, está configurado para realizar esta acción cada segundo, si se encuentra un cuerpo inmediatamente se envía una señal de alarma para ser registrada y almacenada.

El modelo de sensor ultrasónico instalado es el HC-SR04, el mismo que tiene las siguientes características más relevantes:

Tabla 3-5: Tabla de características sensor HC-SR04

Tipo de sensor	Ultrasónico
Rango de medición	2 cm a 400 cm
Voltaje de alimentación	5 V
Precisión	3mm
Interfaz de comunicación	4 hilos, Vcc, Trigger, Echo, GND
Apertura del pulso ultrasónico	15°

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

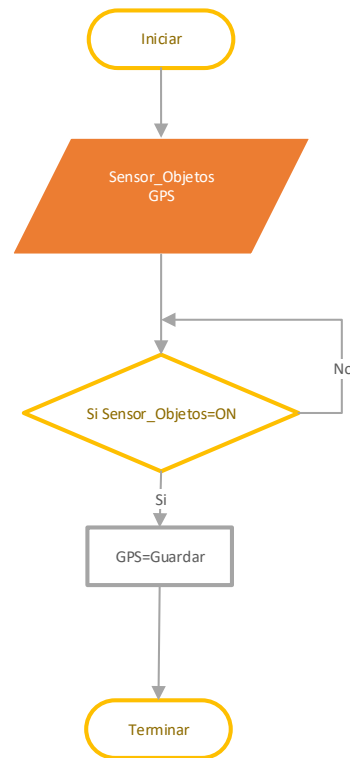


Figura 3-43: Diagrama de flujo de Sensor de Objetos

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.2.6.3. Servicio de posición global (GPS)

El robot cuenta además con módulo de localización, que permitirá almacenar la posición de todas las alarmas que se hayan producido, además de saber la posición del propio robot.

El poder conocer la longitud y latitud de una alarma de temperatura o de distancia, permitirá al operador, luego de terminar la inspección, poder realizar una revisión y si fuese el caso un mantenimiento de los puntos exactos en los que se registró un posible problema.

Para ello, se instaló un módulo que se comunica mediante serial con la tarjeta de control Arduino, como se indica en el siguiente diagrama:

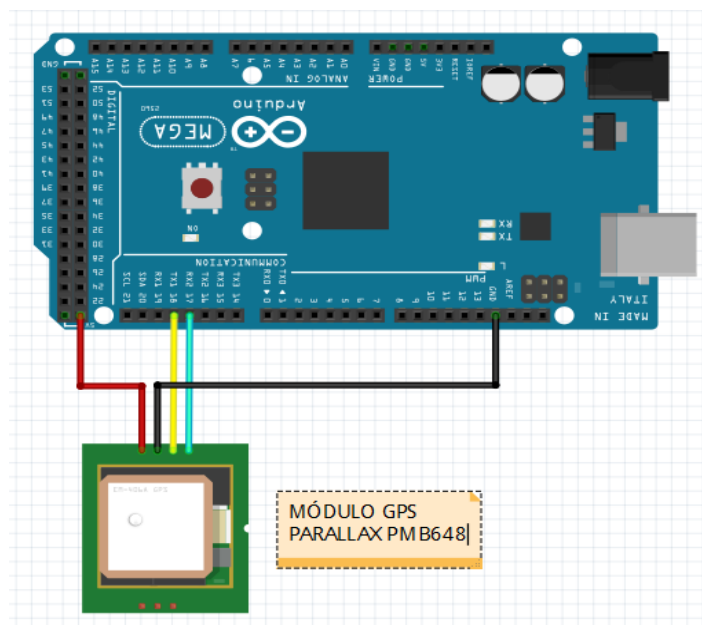


Figura 3-44: Diagrama de conexión módulo GPS

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Se instaló, el modulo GPS Sirf PMB648 de Parallax con una antena integrada, proporciona cadenas estándar raw NMEA0183 (National Marine Electronics Association), rastreando hasta 20 satélites, .El módulo proporciona datos actuales de tiempo, latitud, longitud, velocidad y orientación/dirección de viaje, junto con otros datos y puede usarse en una amplia variedad de aplicaciones comerciales, incluyendo navegación, sistemas de rastreo, mapeo , administración de flotas, auto-piloto y robótica, también se pueden señalar las siguientes características mas relevantes:

Tabla 3-6: Características módulo GPS

Controlador GPS	SiRFstar II
Recepción	Hasta 20 satélites
Voltaje de alimentación	3.3 a 5 VDC
Precisión	2m a 5m
Tiempo de conexión.	1 sec a 42 sec
Comunicación	Serial RS232, TTL
Velocidad de comunicación	4800 bps

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

A continuación, se muestra el diagrama con los componentes anteriormente descritos de manera individual, siendo los más relevantes y de los cuales depende el funcionamiento del robot.

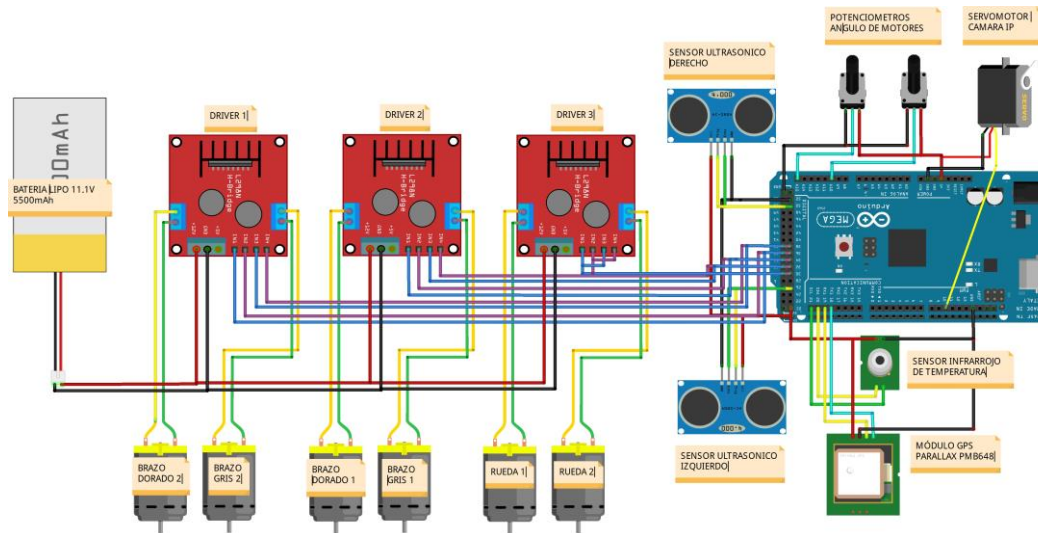


Figura 3-45: Diagrama de los principales componentes electrónicos.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

3.3. Interfaz de usuario

Para la interfaz de usuario se ocupó el software de National Instrument, LabView, debido a sus muchas funciones de bloques que permiten acoplarse a todos los componentes del robot que tienen diversas características. Además de ofrecer la posibilidad de realizar un diseño bastante intuitivo, funcional y sencillo.

La interfaz de usuario dependerá de dos tipos de comunicaciones, la primera que es la comunicación con la cámara ip a bordo del robot, para obtener el video en tiempo real, y la segunda que es la comunicación mediante el módulo Xbee, con todos los componentes del robot.

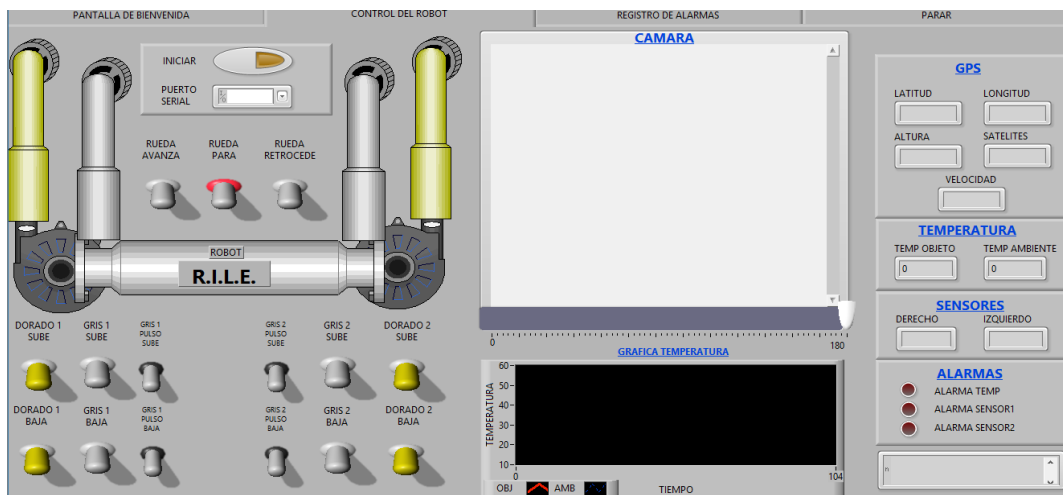


Figura 3-46: Interfaz de usuario.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Mediante la interfaz de usuario el operador tendrá la capacidad de controlar el robot móvil, cada una de sus articulaciones y el desplazamiento del robot, es decir de sus ruedas. Además, en el diseño de la interfaz se podrá acceder a la visualización del video en tiempo real mediante la cámara ip a bordo del robot.

PANTALLA DE BIENVENIDA						
CONTROL AUTOMATICO						
REGISTRO DE ALARMAS						
PARAR						
REGISTRO DE ALARMAS						
FECHA	HORA	SENSOR	TIPO DE ALARMA	VALOR	LATITUD	LONGITUD
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	15.19 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.78 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.50 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.91 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.84 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.78 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.91 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.78 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.78 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.91 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.57 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.78 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.57 cm	0.000000	0.000000
sáb., 27 de may. de 2017	19:50	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	14.43 cm	0.000000	0.000000

Figura 3-47: Registro de alarmas de la interfaz de usuario

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Y finalmente se almacenará la información de los valores de los sensores en el transcurso del recorrido de inspección, con posibles posiciones en donde se requiera un posterior mantenimiento o se requiera alguna potencial intervención.

Además de visualizar el estado de los sensores mientras se realiza la inspección, también se crea un registro con todas las alarmas originadas, las mismas que se pueden visualizar en la siguiente pestaña de la interfaz de usuario, y que una vez que se haya parado el programa, se almacenara en documento de Excel en conjunto con los almacenamientos previos.

K72							
	A	B	C	D	E	F	G
1	FECHA	HORA	SENSOR	TIPO DE ALARMA	VALOR	LATITUD	LONGITUD
56	lun., 22 de may. de 2017	22:21	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	8.50 cm	-1.262048	-78.632530
57	lun., 22 de may. de 2017	22:21	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	8.50 cm	-1.262052	-78.632560
58	lun., 22 de may. de 2017	22:21	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	8.50 cm	-1.262028	-78.632583
59	lun., 22 de may. de 2017	22:22	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	9.81 cm	-1.262033	-78.632591
60	lun., 22 de may. de 2017	22:22	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	2.22 cm	-1.262077	-78.632568
61	lun., 22 de may. de 2017	22:23	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	8.36 cm	-1.262108	-78.632431
62	lun., 22 de may. de 2017	22:24	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	8.50 cm	-1.262038	-78.632514
63	mar., 23 de may. de 2017	17:36	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	3.67 cm	-1.262010	-78.632415
64	mar., 23 de may. de 2017	17:36	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	10.50 cm	-1.261928	-78.632347
65	mar., 23 de may. de 2017	17:36	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	19.19 cm	-1.261928	-78.632347
66	mar., 23 de may. de 2017	17:36	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	17.88 cm	-1.262303	-78.632369
67	mar., 23 de may. de 2017	17:39	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	12.09 cm	-1.262077	-78.632431
68	mar., 23 de may. de 2017	22:46	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	4.22 cm	-1.262215	-78.632057
69	mar., 23 de may. de 2017	22:46	SENSOR ULTRASONICO DE DISTANCIA DERECHO	OBJETO LADO DERECHO	4.57 cm	-1.262195	-78.632072
70	mar., 23 de may. de 2017	22:47	SENSOR ULTRASONICO DE DISTANCIA	OBJETO LADO IZQUIERDO	2.36 cm	-1.262180	-78.632110
71	mar., 23 de may. de 2017	22:47	SENSOR INFRARROJO DE TEMPERATURA	TEMPERATURA DE CABLE ALTA	31.43 °C	-1.262182	-78.632110
72	mar., 23 de may. de 2017	22:47	SENSOR INFRARROJO DE TEMPERATURA	TEMPERATURA DE CABLE ALTA	34.85 °C	-1.262182	-78.632110

Figura 3-48: Registro alarmas almacenadas en Excel.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

Una vez realizado una investigación previa de los problemas y necesidades existentes en la inspección de líneas eléctricas, se ha podido llegar al diseño de una posible solución, por lo que a continuación se detallan los resultados obtenidos en sus pruebas de funcionamiento.

Para poder realizar un análisis de funcionamiento y sus resultados, es conveniente realizar una comparación del diseño del robot propuesto con el resultado final del robot físicamente implementado.

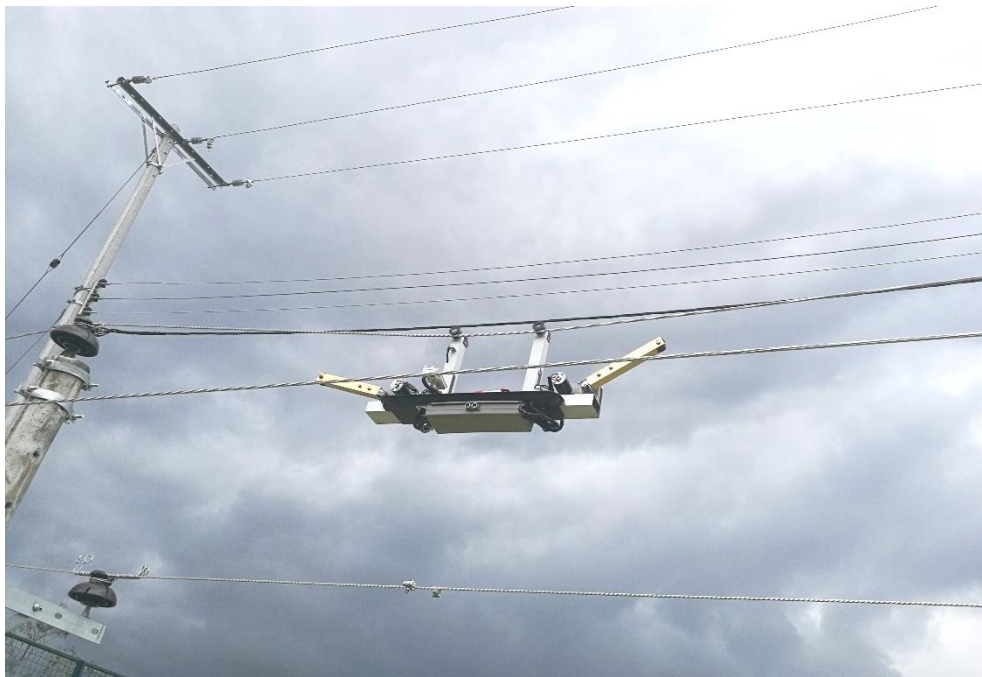


Figura 4-1: Robot de inspección sobre el cableado eléctrico.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Además, se realizará comparaciones entre el sistema propuesto con métodos de inspección realizados de manera manual por un operador y sistemas similares para inspeccionar líneas eléctricas, con esto se busca poder analizar ventajas y desventajas de

cada método, y las mejores condiciones o innovaciones que brinda el robot para inspección de líneas eléctricas.

4.1. Funcionamiento mecánico

Para la construcción del robot se utilizó aluminio como material para la base y los brazos, además de uniones y accesorios de acero para los acoples entre los motores y los brazos. El funcionamiento mecánico de los brazos y las ruedas del robot entregan movimientos como los que se simularon en la etapa de diseño.



Figura 4-2: Construcción de partes mecánicas y acoples.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Cabe mencionar que, en el diseño y simulación del robot ideal, a los resultados finales del funcionamiento mecánico se agregan ciertas imprecisiones a nivel de piñones o acoples, que no son notorias, pero pueden afectar al funcionamiento con exactitud del robot. Esto puede mejorarse utilizando motores de mayor precisión y gran torque, que fueron descartados de adquirirse por su elevado precio.

Una de las principales habilidades del funcionamiento mecánico es poder desplazarse sobre el cableado eléctrico y poder superar los obstáculos que se encuentren en el recorrido, como se muestra a continuación:

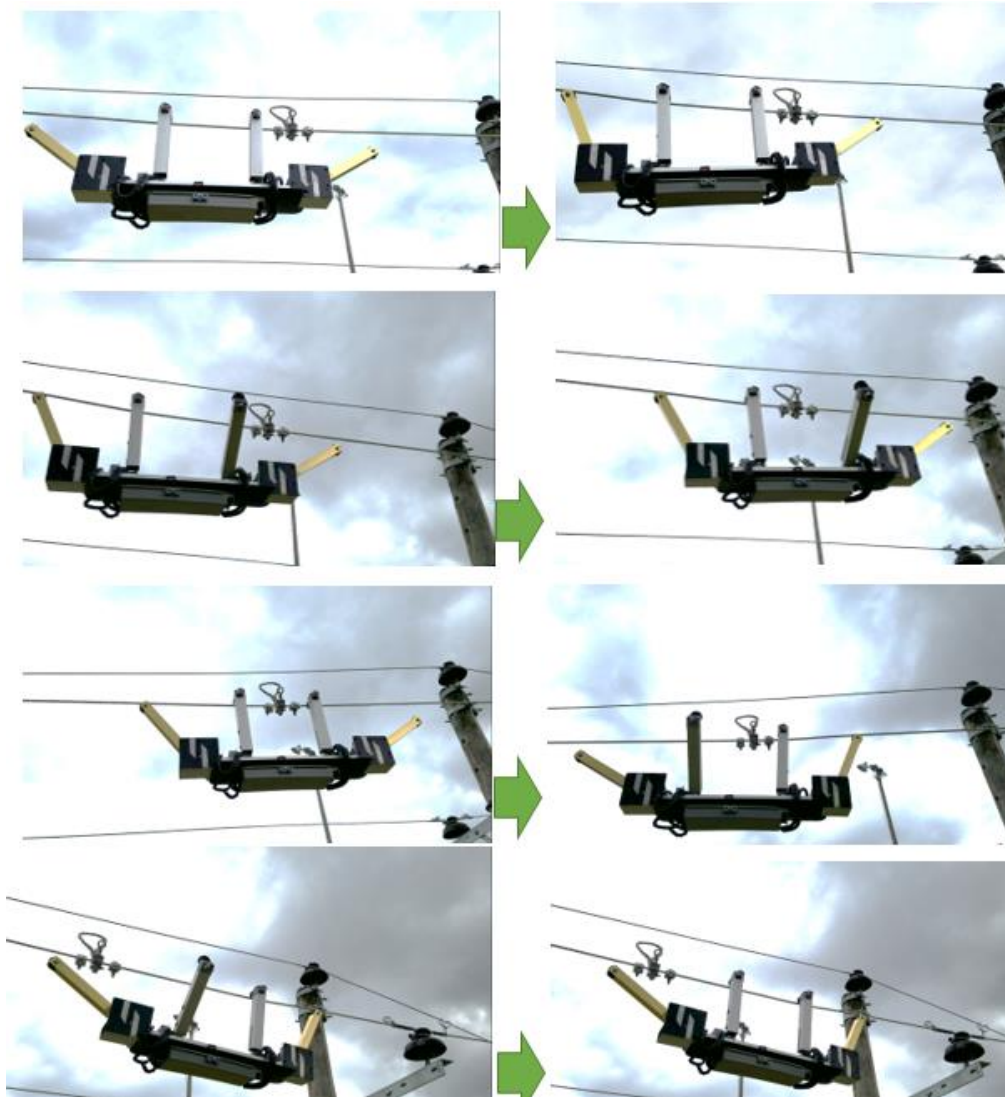


Figura 4-3: Movimientos del robot para superar un obstáculo en el cableado.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Además, es importante destacar que, en las pruebas de funcionamiento, el robot se desplazó en cableados de hasta 30 grados de inclinación, sin ningún problema de movilidad.



Figura 4-4: Robot movilizand en cableado recto e inclinado.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

En las pruebas de funcionamiento se pudo observar que el viento o las vibraciones en el cableado al que se encuentra sujeto el robot, pueden generar que este se balancee, haciendo eje en sus brazos internos (grises), pero la sujeción del robot no es alterada, por lo que no existe ningún tipo de problema en el que el robot pueda caer o soltarse del cableado eléctrico.

4.2. Funcionamiento de la comunicación del robot

Los resultados en la comunicación en general entre el robot y el operador, funcionan en los rangos esperados de acuerdo al diseño, además de las características de funcionamiento de los Xbee, mismo que se realizaron pruebas en áreas abiertas de hasta 60m de distancia, teniendo un funcionamiento correcto tanto en los datos enviados como en los recibidos.

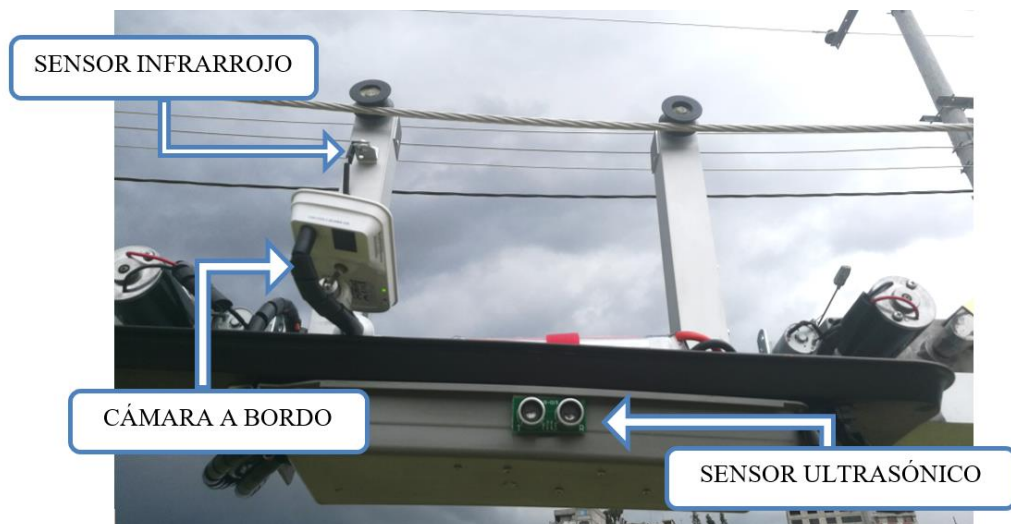


Figura 4-5: Ubicación de sensores y cámara a bordo.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

El funcionamiento de la cámara, es vía wifi, el mismo que depende directamente del tipo de antena del pc, como de la cámara, los mismo que en espacios abiertos tuvieron un funcionamiento correcto de hasta 30m aproximadamente. En el caso de querer ampliar la distancia de cobertura, se debería instalar una antena externa de mayor cobertura, pero el cambio solo es a nivel de hardware.

4.3. Funcionamiento de los componentes electrónicos.

Todos los dispositivos instalados, trabajaron de manera fluida y correcta en pruebas de hasta 1h de funcionamiento con el sistema encendido y alimentado mediante la batería.

Las medidas de los sensores de distancia, entregan medidas de hasta 3m con una precisión y errores de medición de aproximadamente 5cm, por los que los valores son muy confiables, para esta aplicación en la cual no se necesita valores con gran exactitud.

El sensor de temperatura trabaja de tal manera que se puede visualizar los valores en tiempo real sin inconvenientes, además de poder ocuparse de cambios de temperatura muy abruptos, lo que a la mayoría de sensores de temperatura del mercado cuesta mucho realizar este tipo de mediciones con rápidas variaciones.

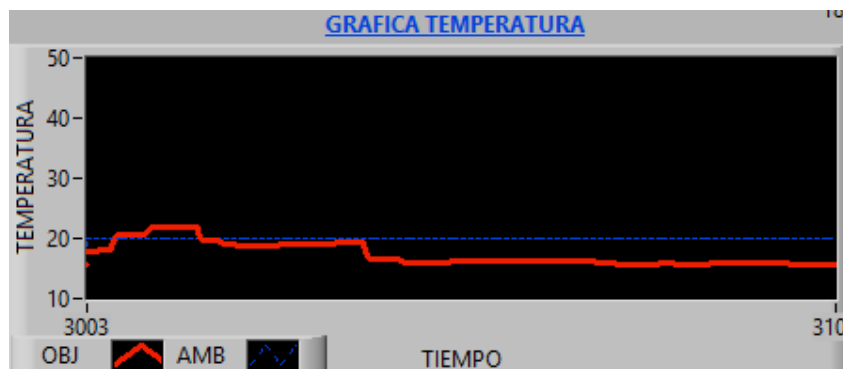


Figura 4-6: Gráfica de valores de temperatura del cable y ambiente.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

Los drivers de los motores son capaces de controlar los movimientos de los mismos, pero de manera individual, es decir no más de un motor a la vez, ya que la carga podría sobrepasar la alimentación que puede entregar la batería. Esto no es inconveniente dado que el operador manipula los brazos de robot, paso por paso, para cumplir con las secuencias de movimientos.



Figura 4-7: Imagen de cámara a bordo controlada por servomotor.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

El servomotor que entrega 180 grados de movilidad a la cámara ip, permite poder visualizar los movimientos del robot, así como poder ir inspeccionando el estado del cableado eléctrico y sus accesorios.

4.4. Funcionamiento de la interfaz de usuario

La interfaz de usuario permite le permite al operador trabajar en un entorno muy intuitivo, la mayoría de las personas que usaron el HMI, indicaron que su uso es sencillo y que no cuesta acostumbrarse.

El registro de los valores de tipo alarma son almacenados en la tercera pestaña de la interfaz de usuario, lo que permite ir visualizando las alarmas de los sensores mientras se realiza la inspección.

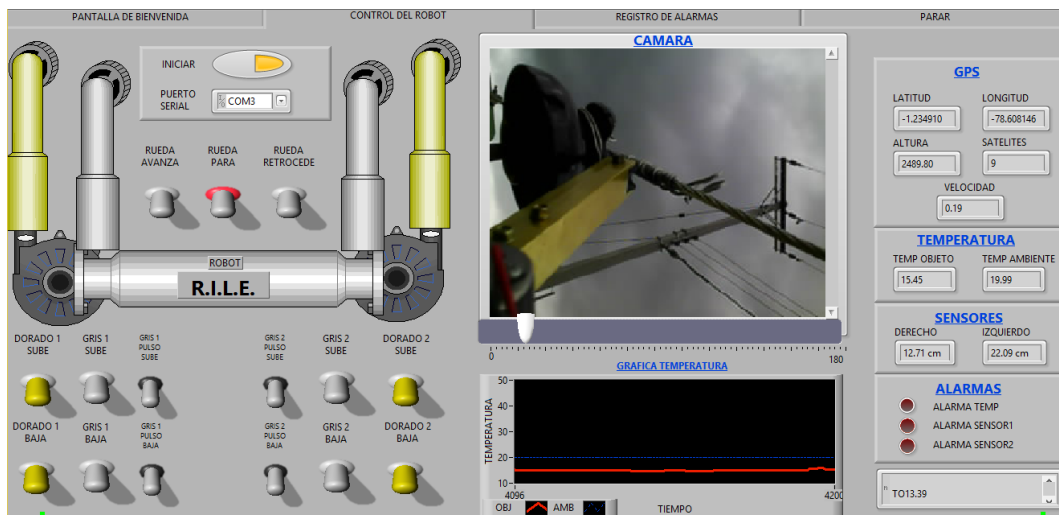


Figura 4-8: Interfaz de usuario con datos de funcionamiento.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

El software LabVIEW permite crear un archivo ejecutable para la interfaz de usuario, con lo que se puede evitar la necesidad de adquirir una licencia completa, si una entidad pública o privada lo usara.

4.5. Funcionamiento del robot en campo.

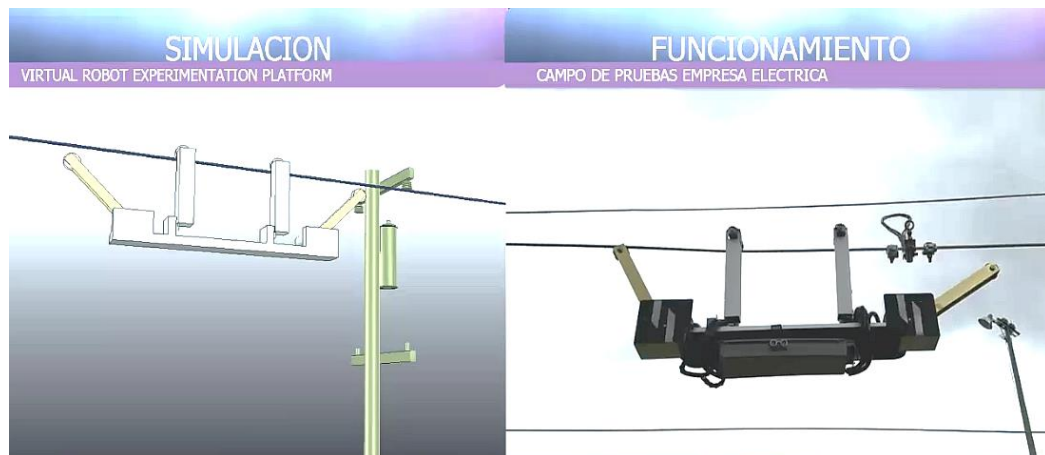


Figura 4-9: Simulación y funcionamiento del robot.

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

El funcionamiento del robot se realizó en el complejo de campo de pruebas de la Empresa Eléctrica de Ambato, por motivos de seguridad se desenergizó los puntos donde se ejecutaron los ensayos con el robot. Se efectuaron sus principales movimientos, desplazamientos control a distancia y demás, sin problemas, pudiendo superar obstáculos en la línea eléctrica como se mostraron en la Fig.73.

Tabla 4-1 Análisis del robot de inspección sobre métodos tradicionales

Características	Robot de inspección de líneas eléctricas	Métodos tradicionales realizador por personal humano
Inspección de vegetación y objetos cercanos a las líneas eléctricas	La medición se realiza desde la líneas eléctricas lo que permite un mejor análisis	La medición se realiza desde bajo la línea eléctrica y en muchos casos es de difícil acceso.
Inspección de puntos calientes	No solo se revisan los puntos calientes, sino la temperatura a lo largo de todo el cableado	Se revisan los puntos calientes desde puntos lejanos, lo que puede requerir otra inspección para poder determinar su causa
Inspección visual	La cámara a bordo del robot permite ir visualizando el cableado eléctrico a lo largo	Si la inspección se realiza a larga distancia por binoculares, no se

	de todo el recorrido, así como también de sus accesorios, y todo esto a una distancia menor a 50cm	podrá realizar una observación eficiente. Si la inspección se realiza sobre el cableado eléctrico, el trabajador está expuesto a constante riesgo.
Registro de inspecciones	El robot va almacenando una tabla digital con registros que contienen el lugar y tiempo en el que sucedió cualquier alarma.	El operador al estar sobre la línea eléctrica no tiene la capacidad de ir anotando de manera digital ni manual cualquier evento visualizado
Riesgo	El robot está en continuo contacto con el cableado eléctrico que se encuentra alimentado. Pero cualquier inconveniente solo produciría daños materiales	El trabajador, aunque se encuentre con el equipo y protección necesaria, estará siempre expuesto a riesgo eléctrico y también corre el riesgo de una caída de gran altura. Por lo que en el peor de los casos está en riesgo una vida humana.
Tiempo	El tiempo estimado en recorrer un vano eléctrico de 100m es de 7 minutos	El tiempo estimado en recorrer un vano de 100m es de 5 minutos

Fuente: EIBAR BEJARANO, Carlos Gabriel, 2017

CONCLUSIONES

Ya que el robot se controla de forma remota, es decir teleoperado, primordialmente se evitó que el operador trabaje sobre las líneas eléctricas, con lo cual se logra disminuir el riesgo cerca de su totalidad, al que el o los trabajadores están expuestos al inspeccionar las líneas eléctricas con métodos tradicionales, evitando que el operador pueda sufrir accidentes y resguardando lo más importante, su vida.

De acuerdo a las investigaciones de robots usados en la inspección de líneas eléctricas, la mayor complicación es poder superar los obstáculos. Enfocado en este aspecto se puede aseverar que el robot tiene la capacidad de realizar todas las tareas que se plantearon como objetivos, una de las cuales era el poder superar obstáculos, por lo que se concluye que, el diseño previo aseguró una implementación acertada y funcional del robot, permitiéndole realizar su trabajo de manera adecuada.

En lo referente a los beneficios que aporta el robot de inspección de líneas eléctricas frente a los métodos tradicionales de inspección, se puede concluir primero que el llevar registros detallados de las inspecciones presenta muchas ventajas ya que, al obtener estos datos de manera digital, se puede realizar un análisis mediante graficas u otros métodos, que posteriormente permitan realizar los mantenimientos de manera ordenada y planificada, además de poder realizar un análisis para en un futuro detectar en donde se producen mayores casos de incidentes en las líneas eléctricas.

Las inspecciones se realizan de manera más eficiente ya que el robot permite ir realizando tres tipos de inspecciones de manera simultánea, lo cual con los métodos tradicionales se realizan de forma separada, por lo que disminuye los trabajos de inspección en un 66% en lo referente a la cantidad de trabajo realizado.

La estructura del robot permite seguir trabajando y mejorando el diseño actualmente funcional, principalmente adicionar otro campo de acción, es decir que el robot permita realizar trabajos de mantenimiento de nivel básico además de los trabajos de inspección, esto incrementaría la eficiencia en los trabajos, además de disminuir riesgos al realizar estos tipos de mantenimiento de manera manual.

RECOMENDACIONES

Los trabajos de inspección son realizados a la intemperie, por lo que se recomienda realizar dichas inspecciones con un clima favorable, principalmente sin lluvias, ya que varios de los elementos del robot no tienen la suficiente impermeabilidad para trabajos en lluvia.

El robot y sus componentes electrónicos, son alimentados mediante una batería la cual puede disminuir su vida útil si no es cargada de forma adecuada, por lo que se recomienda de preferencia realizar la carga antes de que el porcentaje de batería llegue a menos del 30%.

El uso de la interfaz de usuario a pesar de ser muy intuitivo requiere un tiempo para familiarizarse, para esto se recomienda realizar pruebas del funcionamiento del robot antes de poner en funcionamiento sobre las líneas eléctricas.

Al realizar el control del robot, lo mejor es realizar una tarea a la vez, tanto por el consumo que requieren como también por el control adecuado del operador, ya que se necesita controlar la cámara a bordo de manera continua para poder explorar el desplazamiento del robot, lo cual al realizarse de manera simultánea puede provocar un incorrecto control de las ruedas del robot, por lo que se recomienda parar las ruedas del robot antes de realizar cualquier otra tarea.

BIBLIOGRAFÍA

AIN, REE. (2004). *Proyecto Pelicano: Robot aéreo para inspección de líneas eléctricas*. Madrid: Hisparob, Plataforma Tecnológica Española de Robótica .

ARDUINO. (13 de Febrero de 2017). *Arduino project*. Obtenido de www.arduino.cc

ARMENDÁRIZ, M. I. (20 de Octubre de 2016). Riesgos en el mantenimiento de instalaciones de alta tensión en tensión y la aplicación del RD 614/2001. Navarra, España: UPNA.

ARMESTO, L. (18 de 08 de 2016). *Introducción a V-REP en Brazo Articulado*. Obtenido de robotica.webs.upv

BARAJAS, O. (2010). Arman robot inspector de líneas eléctricas . *Electrónicos online*.

CHECA, L. M. (2006). *Lineas de Transporte de Energía*. Barcelona: S.A. Marcombo.

COPPELIAROBOTICS. (29 de Agosto de 2016). *V-REP Create, compose, simulate, any robot*. Obtenido de www.coppeliarobotics.com

DEBENEST, P. (2008). Expliner – Robot for Inspection of Transmission Lines. *IEEE International Conference*.

- ELIZONDO, D., GENTILE, T., & CANDIA, H. (2010). Overview of Robotic Applications for Energized Transmission Line Work –Technologies, Field Projects and Future Developments. *1st International Conference on Applied Robotics for the Power*.
- GONÇALVES, R. S. (2013). Review and latest trends in mobile robots used on power transmission lines. *International Journal of Advanced Robotic Systems*, 10(12), 408.
- LEÓN RIVAS, M. B. (2013). Estudio de soluciones existentes en el mercado para la inspección y mantenimiento de líneas eléctricas de alta tensión. *Digital UPM*.
- LLAMAS, L. (22 de Octubre de 2016). *El bus i2c en arduino*. Obtenido de <https://www.luisllamas.es/arduino-i2c/>
- LLAMAS, LUIS. (10 de Diciembre de 2016). Obtenido de Medir distancia con Arduino y sensor de ultrasonidos HC-SR04: <https://www.luisllamas.es>
- MCI ELECTRONICS. (25 de Noviembre de 2016). *XBee*. Obtenido de Ingeniería MCI Ltda: <http://xbec.cl/que-es-xbec/>
- MELEXIS. (2009). *MLX90614 family*. Belgium: Microelectronic Integrated Systems.
- NATIONAL INSTRUMENTS CORPORATION. (2016). *National Instruments*. Obtenido de <http://sine.ni.com/nips/cds/view/p/lang/es/nid/212478>

NICOLAS POULIOT, P.-L. R. (2015). LineScout Technology Opens the Way to Robotic Inspection and Maintenance of High-Voltage Power Lines. *IEEE*.

PARALLAX INC. (2011). *GPS Parallax Manual*. California: Parallax Inc.

PROMETEC. (07 de Enero de 2017). *El bus i2c*. Obtenido de <http://www.promotec.net>

R. ARACIL, L. P. (1995). ROBTET: A new teleoperated system for live-line. *Proceedings of the 7th International Conference on Transmission and Distribution Construction and Live Line Maintenance*, 205-211.

ROGÉRIO SALES GONÇALVES, J. C. (2013). Review and Latest Trends in Mobile Robots Used on Power Transmission Lines. *International Journal of Advanced Robotic Systems*.

SIRF TECHNOLOGY. (2005). *NMEA Reference Manual*. New Zeland: Navman.

WANG WEI, B. Y. (2013). The Mechanism of a Snake-Like Robot's Clamping Obstacle Navigation on High Voltage Transmission Lines. *International Journal of Advanced Robotic Systems*.

ANEXOS

PRESUPUESTO

Costos directos:

Materiales	Cantidad	Precio Unitario	Precio Cantidad
Cámaras inalámbricas IP	1	200	200
Arduino Mega	1	50	50
Módulo Xbee	2	100	200
Shield Xbee	1	30	30
Dongle USB Xbee	1	30	30
Sensores Ultrasónicos	2	30	60
Sensor Infrarrojo	1	50	50
Servomotor	1	20	20
Motores con caja Reductora	4	150	600
Piezas impresas en 3D	1	100	100
Batería	1	80	80
Ruedas tipo polea	4	50	200
Modulo GPS	1	80	80
Total Costos Directos			\$ 1680

Costos indirectos:

Gasto		Precio
Transporte		\$ 50.00
Impresiones, copias, internet		\$ 100.00
Otros		\$ 100.00
Total	Costos Indirectos	\$ 250.00

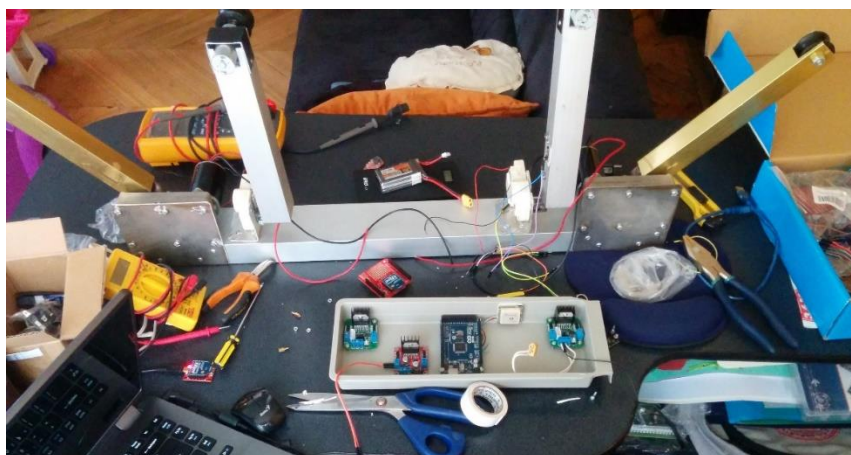
FOTOGRAFÍAS:



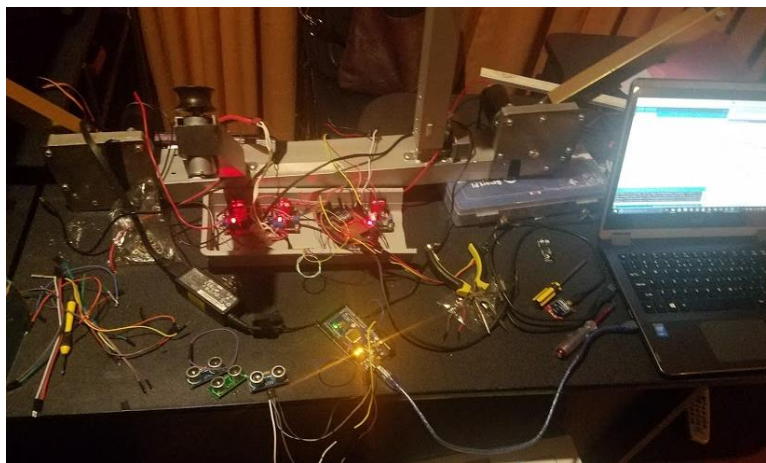
Fotografía 1



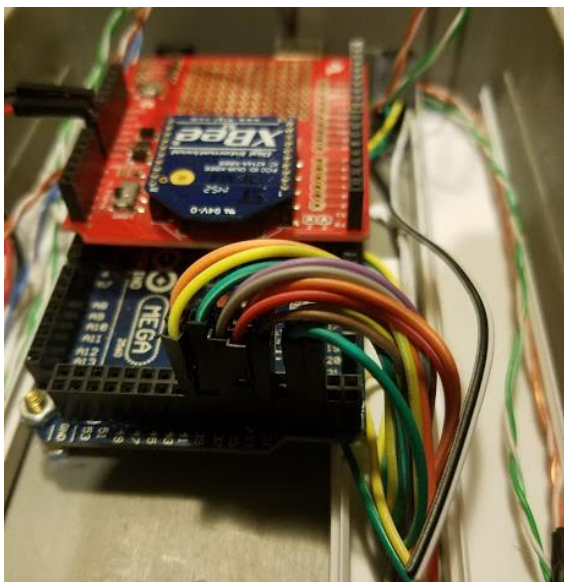
Fotografía 2



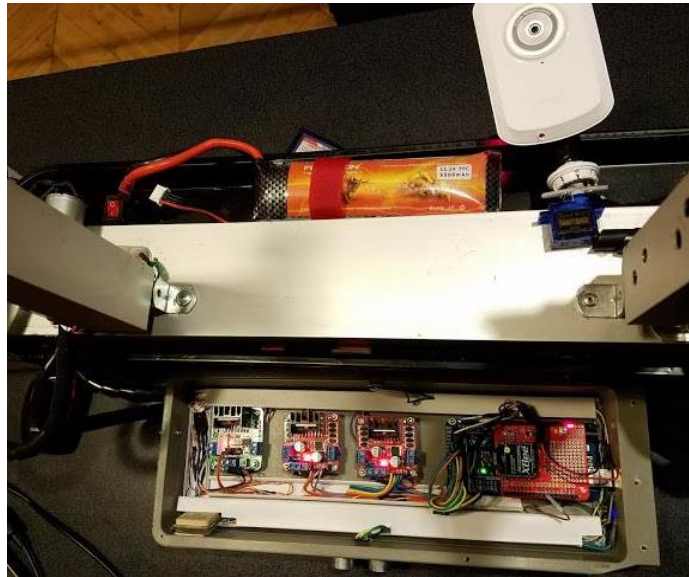
Fotografía 3



Fotografía 4



Fotografía 5



Fotografía 6



Fotografía 7



Fotografía 8



Fotografía 9



Fotografia 10



Fotografia 11

CÓDIGO ARDUINO

```
//-----LIBRERÍAS-----
#include <Wire.h>           // Libreria de comunicacion I2C
#include <SparkFunMLX90614.h> // Libreria de sensor MLX90614
#include <TinyGPS.h>        // Libreria de GPS
#include <HCSR04.h>         // Libreria Sensor Ultrasonico
#include <Servo.h>          // Libreria Servo de camara

//-----CONSTANTES Y VARIABLES -----
unsigned long intervalo=500; // tiempo que espera los sensores ultrasonicos en ms
unsigned long MillisAnterior=0; // millis() retorna una variable tipo long unsigned.
unsigned long MillisAnterior2=0; // millis() retorna una variable tipo long unsigned.
unsigned long MillisAnteriorGPS=30000; // millis() retorna una variable tipo long unsigned.
unsigned long MillisAnteriorGPS1=1000; // millis() retorna una variable tipo long unsigned.
unsigned long MillisAnteriorTEMP=4000; // millis() retorna una variable tipo long unsigned.
// Variables Servo
int val = 90; //Variable de entrada del Serial
int pos = 90; //Variable movimiento servo
Servo servo; //objeto Servo
//sensor de distancia 1
HCSR04 Sensor1(26,27); //26trig 27Echo
float distancia1= 0;
float maxdistancia1=30;
//sensor de distancia 2
HCSR04 Sensor2(52,53); //52trig 53Echo
float distancia2= 0;
float maxdistancia2=30;
//SENSOR IR de temperatura
float maxTO=30;
float TO;
float TA;

IRTherm therm; // INICIA VARIABLE DE LIBRERIA DE TEMPERATURA
TinyGPS gps; // INICIA VARIABLE DE LIBRERIA DE GPS

void setup()
{
  Serial.begin(9600); //XBEE
  Serial1.begin(4800); //GPS
  SetGPSSWalkMode(); // CONFIGURA GPS EN MODO CAMINATA
  therm.begin(); // INICIA EL SENSOR DE TEMPERATURA
  therm.setUnit(TEMP_C); // CONFIGURA LA LIBRERIA EN CENTIGRADOS
  pinMode(10, OUTPUT); //LED DE SENSOR1
  pinMode(11, OUTPUT); //LED DE SENSOR2
  servo.attach(9); //SERVO AL PIN DIGITAL 9
  pinMode (A10, INPUT);
  pinMode (A15, INPUT);
  Serial.println("ROBOT RILE INICIANDO");
}
void loop()
{
  if(Serial.available() > 0){ //Comprobar si recibe un byte via Serial
    char mensaje = Serial.read(); //almacenar byte en la variable mensaje

    //Si el mensaje es G REGRESA LOS DATOS DE GPS.
    if(mensaje == char('S')){
      SERVO();
    }

    //Si el mensaje es G REGRESA LOS DATOS DE GPS.
    if(mensaje == char('G')){
      GPS();
    }

    //Si el mensaje es Z REGRESA LOS DATOS DE GPS PARA ALARMA.
    if(mensaje == char('Z')){
      GPS2();
    }

    //Si el mensaje es T REGRESA LOS DATOS DE TEMPERATURA.
    if(mensaje == char('T')){
      TEMPERATURA();
    }

    //Si el mensaje es M ACTIVA RUEDAS DE AVANCE.
```

```

if(mensaje == char('M')){
  AVANZAR();
}
//Si el mensaje es N ACTIVA RUEDAS DE RETROCEDER.
if(mensaje == char('N')){
  RETROCEDER();
}
//Si el mensaje es P PARA LAS RUEDAS .
if(mensaje == char('P')){
  PARAR();
}
//Si el mensaje es A ACTIVA EL GRIS1 ARRIBA.
if(mensaje == char('A')){
  SUBIRG1();
}
//Si el mensaje es B ACTIVA EL GRIS1 ABAJO.
if(mensaje == char('B')){
  BAJARG1();
}
//Si el mensaje es a ACTIVA EL GRIS1 ARRIBA POR PULSOS.
if(mensaje == char('a')){
  SUBG1();
}
//Si el mensaje es b ACTIVA EL GRIS1 ABAJO POR PULSOS.
if(mensaje == char('b')){
  BAJG1();
}

//Si el mensaje es C ACTIVA EL GRIS2 ARRIBA.
if(mensaje == char('C')){
  SUBIRG2();
}

//Si el mensaje es D ACTIVA EL GRIS2 ABAJO.
if(mensaje == char('D')){
  BAJARG2();
}
//Si el mensaje es a ACTIVA EL GRIS1 ARRIBA POR PULSOS.
if(mensaje == char('c')){
  SUBG2();
}
//Si el mensaje es b ACTIVA EL GRIS1 ABAJO POR PULSOS.
if(mensaje == char('d')){
  BAJG2();
}

//Si el mensaje es E ACTIVA EL DORADO1 ARRIBA.
if(mensaje == char('E')){
  SUBIRD1();
}
//Si el mensaje es F ACTIVA EL DORADO1 ABAJO.
if(mensaje == char('F')){
  BAJARD1();
}
//Si el mensaje es H ACTIVA EL DORADO1 ARRIBA.
if(mensaje == char('H')){
  SUBIRD2();
}
//Si el mensaje es I ACTIVA EL DORADO1 ABAJO.
if(mensaje == char('I')){
  BAJARD2();
}
}
SENSOR1();
SENSOR2();
GPS();
TEMPERATURA();
}

//-----FUNCION SERVO-----
void SERVO(){

if(Serial.available() > 0) //Detecta si hay alguna entrada por serial
{
  pos = Serial.parseInt();
  //Serial.print(pos,DEC);

```

```

}

while(pos < val)
{
  //Serial.println(val);
  val--;
  servo.write(val);
  delay(20);
}
while(pos > val)
{
  //Serial.println(val);
  val++;
  servo.write(val);
  delay(20);
}
}

//-----FUNCION GPS-----
void GPS(){
  bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;

  //-----INICIA GPS-----
  // Se envia por 1 segundo valores de GPS
  for (unsigned long start = millis(); millis() - start < 1000;)
  {
    while (Serial1.available())
    {
      char c = Serial1.read();
      //Serial.write(c); // VER EL FLUJO DE DATOS DEL GPS
      if (gps.encode(c)) // LEE SI HAY UN NUEVO DATO DEL GPS
        newData = true;
    }
  }
  if (newData)
  {
    float flat, flon;
    unsigned long age;
    unsigned long MillisActualGPS = millis(); // almacena el tiempo actual
    if ((unsigned long)(MillisActualGPS - MillisAnteriorGPS) >= 30000) {
      gps.f_get_position(&flat, &flon, &age);
      Serial.print(" LA");
      Serial.println(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
      Serial.print(" LO");
      Serial.println(flou == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
      Serial.print(" SA");
      Serial.println(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 : gps.satellites());
      Serial.print(" AL");
      Serial.println(gps.f_altitude());
      Serial.print(" PR");
      Serial.println(gps.hdop() == TinyGPS::GPS_INVALID_HDOP ? 0 : gps.hdop());
      MillisAnteriorGPS= millis();
    }
    unsigned long MillisActualGPS1 = millis(); // almacena el tiempo actual
    if ((unsigned long)(MillisActualGPS1 - MillisAnteriorGPS1) >= 5000) {
      Serial.print(" VE");
      Serial.println(gps.f_speed_kmph());
      MillisAnteriorGPS1= millis();
    }
  }
}

//-----FUNCION GPS SIN TIEMPO-----
void GPS2(){
  float flat, flon;
  unsigned long age;
  gps.f_get_position(&flat, &flon, &age);
  Serial.print(" LA");
  Serial.println(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
  Serial.print(" LO");
  Serial.println(flou == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
}

```

```

//-----FUNCION TEMPERATURA-----
void TEMPERATURA(){
//.....INICIA I2C TEMP
  unsigned long MillisActualTEMP = millis(); // almacena el tiempo actual

  if (therm.read())
  {

    TO= therm.object();
    TA= therm.ambient();

    Serial.println();
    Serial.print(" TO"); //TEMPERATURA DEL OBJETO EN CENT
    Serial.println(TO);

    if(TO>maxTO){
      digitalWrite(43, HIGH);
      GPS2();
      Serial.print(" AT");
      Serial.print(TO);
    }
    else
    {
      digitalWrite(43, LOW);
    }

    if ((unsigned long)(MillisActualTEMP - MillisAnteriorTEMP) >= 4000) {

      Serial.print(" TA"); //TEMPERATURA DEL AMBIENTE EN CENT
      Serial.println(TA);

    }
    MillisAnteriorTEMP= millis();
  }
}
//.....TERMINA I2C TEMP

//-----SENSOR 1-----

void SENSOR1(){
  unsigned long MillisActual = millis(); // almacena el tiempo actual
  // chequea si el "intervalo" de tiempo ha pasado
  if ((unsigned long)(MillisActual - MillisAnterior ) >= intervalo) {
    //funcionamiento del sensor envia y recibe la señal

    distancia1 = Sensor1.ping(58,615); // Formula: uS / 58 = centimeters ; 500 = offset HC-SR0; 615 = offset US-015

    if(distancia1>0){
      if(distancia1>maxdistancia1){
        digitalWrite(40, LOW);
      }
      if(distancia1<maxdistancia1){
        digitalWrite(40, HIGH);
        GPS2();
        Serial.print(" AS");
        Serial.print(distancia1);
      }
    }
    MillisAnterior= millis();
  }
}

//-----SENSOR 2-----

void SENSOR2(){

  unsigned long MillisActual2 = millis(); // almacena el tiempo actual

  // chequea si el "intervalo" de tiempo ha pasado

  if ((unsigned long)(MillisActual2 - MillisAnterior2 ) >= intervalo) {
    //funcionamiento del sensor envia y recibe la señal
    distancia2 = Sensor2.ping(58,615); // Formula: uS / 58 = centimeters ; 500 = offset HC-SR0; 615 = offset US-015
    if(distancia2>0){
      if(distancia2>maxdistancia2){
        digitalWrite(41, LOW);
      }
    }
  }
}

```

```

if(distancia2<maxdistancia2){
    digitalWrite(41, HIGH);
    GPS2();
    Serial.print(" BS");
    Serial.print(distancia2);
}
}
MillisAnterior2= millis();
}
}
//-----SUBIR GRIS 1-----
void SUBIRG1(){
int G1A=781;          //VALOR DE GRIS1 ALTO 1003
int a1 = analogRead(A15); //VALOR ANALOGICO DE GRIS1

if(a1<G1A){
    while (a1<G1A)
    {
        a1 = analogRead(A15);
        //Serial.println(a1);
        digitalWrite(38,HIGH);
        digitalWrite(39,LOW); // HIGH
        delay(10);
    }

}

digitalWrite(38,LOW);
digitalWrite(39,LOW);
}
//-----BAJAR GRIS 1-----
void BAJARG1(){
int G1B=585;          //VALOR DE GRIS1 BAJO
int a1 = analogRead(A15); //VALOR ANALOGICO DE GRIS1

if(a1>G1B){
    while (a1>G1B)
    {
        a1 = analogRead(A15);
        //Serial.println(a1);
        digitalWrite(38,LOW);
        digitalWrite(39,HIGH); // HIGH
        delay(10);
    }

}

digitalWrite(38,LOW);
digitalWrite(39,LOW);
}

//-----SUBIR GRIS 2-----
void SUBIRG2(){
int G2A=539;          //VALOR DE GRIS2 ALTO 298
int a2 = analogRead(A10); //VALOR ANALOGICO DE GRIS2

if(a2>G2A){
    //Serial.print("BAJO");

    while (a2>G2A)
    {
        digitalWrite(32,HIGH);
        digitalWrite(33,LOW);
        a2 = analogRead(10);
        //Serial.println(a2);
        delay(10);
    }

}

digitalWrite(32,LOW);
digitalWrite(33,LOW);
}

//-----BAJAR GRIS 2-----
void BAJARG2(){
int G2B=715;          //VALOR DE GRIS2 BAJO
int a2 = analogRead(A10); //VALOR ANALOGICO DE GRIS1
int ultimo;

```

```

    if(a2<G2B){
        while (a2<G2B)
        {
            a2 = analogRead(A10);
            //Serial.println(a2);
            digitalWrite(32,LOW);
            digitalWrite(33,HIGH);
            delay(10);
        }
    }
    digitalWrite(32,LOW);
    digitalWrite(33,LOW);
}

//-----SUBIR GRIS 1 POR PULSOS-----
void SUBG1(){

    digitalWrite(39,LOW);
    digitalWrite(38,HIGH);
    delay(30);
    digitalWrite(38,LOW);
    digitalWrite(39,LOW);

}

//-----BAJAR GRIS 1 POR PULSOS-----
void BAJG1(){

    digitalWrite(38,LOW);
    digitalWrite(39,HIGH);
    delay(30);
    digitalWrite(38,LOW);
    digitalWrite(39,LOW);

}

//-----SUBIR GRIS 2 POR PULSOS-----
void SUBG2(){

    digitalWrite(33,LOW);
    digitalWrite(32,HIGH);
    delay(30);
    digitalWrite(32,LOW);
    digitalWrite(33,LOW);

}

//-----BAJAR GRIS 2 POR PULSOS-----
void BAJG2(){

    digitalWrite(32,LOW);
    digitalWrite(33,HIGH);
    delay(30);
    digitalWrite(32,LOW);
    digitalWrite(33,LOW);

}

//-----SUBIR DORADO 1-----
void SUBIRD1(){

    digitalWrite(37,LOW);
    digitalWrite(36,HIGH);
    delay(100);
    digitalWrite(38,LOW);
    digitalWrite(36,LOW);

}

//-----BAJAR DORADO 1-----
void BAJARD1(){

    digitalWrite(36,LOW);
    digitalWrite(37,HIGH);
    delay(100);
    digitalWrite(36,LOW);
    digitalWrite(37,LOW);
}

```

```

}

//-----SUBIR DORADO 2-----
void SUBIRD2(){

    digitalWrite(35,HIGH);
    digitalWrite(34,LOW);
    delay(100);
    digitalWrite(34,LOW);
    digitalWrite(35,LOW);

}

//-----BAJAR DORADO 2-----
void BAJARD2(){

    digitalWrite(34,HIGH);
    digitalWrite(35,LOW);
    delay(100);
    digitalWrite(34,LOW);
    digitalWrite(35,LOW);

}

//-----AVANZAR-----
void AVANZAR(){
    digitalWrite(30,LOW);
    digitalWrite(31,HIGH);
    delay(20);
}

//-----PARAR-----
void PARAR(){
    digitalWrite(30,LOW);
    digitalWrite(31,LOW);
    delay(20);
}

//-----RETROCEDER-----
void RETROCEDER(){
    digitalWrite(30,HIGH);
    digitalWrite(31,LOW);
    delay(20);
}

void SetGPSWalkMode()
{
    // CODIGO DEL PROVEEDOR  CONFIGURA EL GPS EN MODO CAMINATA
    // Band = 4800
    // DataBits = 8
    // StopBits = 1
    // Parity = 0
    // Checksum *0C
    Serial1.println("$PSRF100,0,4800,8,1,0*0F"); //CONFIGURA EL GPS EN MODO BINARIO

    delay(1000);
    Serial1.write((byte)0xA0); // Inicia sequencia
    Serial1.write((byte)0xA2);
    Serial1.write((byte)0x00);
    Serial1.write((byte)0x02);

    Serial1.write((byte)0x8F);
    Serial1.write((byte)0x00);

    Serial1.write((byte)0x00);
    Serial1.write((byte)0x8F);
    Serial1.write((byte)0xB0);
    Serial1.write((byte)0xB3);

    delay(100);

    // //CONFIGURA EL GPS EN MODO NMEA
    Serial1.write((byte)0xA0);
    Serial1.write((byte)0xA2);
    Serial1.write((byte)0x00);
    Serial1.write((byte)0x18);

    delay(100);

    Serial1.write((byte)0x81); // byte 1 Message ID = 129

```

```

Serial1.write((byte)0x02); // byte 2 Mode
Serial1.write((byte)0x01); // byte 3 GGA message (ON) 1 second
Serial1.write((byte)0x01); // byte 4 Checksum
Serial1.write((byte)0x00); // byte 5 GLL message (OFF) 0 second
Serial1.write((byte)0x01); // byte 6 Checksum
Serial1.write((byte)0x00); // byte 7 GSA message (OFF) 0 second
Serial1.write((byte)0x01); // byte 8 Checksum
Serial1.write((byte)0x00); // byte 9 GSV message (OFF) 0 second
Serial1.write((byte)0x01); // byte 10 Checksum
Serial1.write((byte)0x01); // byte 11 RMC message (ON) 1 second
Serial1.write((byte)0x01); // byte 12 Checksum

delay(100);

Serial1.write((byte)0x00); // byte 13 VTG message (OFF) 0 second
Serial1.write((byte)0x01); // byte 14 Checksum
Serial1.write((byte)0x00); // byte 15 MSS message (OFF) 0 second
Serial1.write((byte)0x01); // byte 16 Checksum
Serial1.write((byte)0x00); // byte 17 EPE message
Serial1.write((byte)0x01); // byte 18 Checksum
Serial1.write((byte)0x00); // byte 19 ZDA message
Serial1.write((byte)0x01); // byte 20 Checksum
Serial1.write((byte)0x00); // byte 21 Unused field
Serial1.write((byte)0x01); // byte 22 Unused field
Serial1.write((byte)0x12); // byte 23 Bit rate dec2hex(4800)=12C0
Serial1.write((byte)0xC0); // byte 24

delay(100);

Serial1.write((byte)0x01); // Message Checksum
Serial1.write((byte)0x61);
Serial1.write((byte)0xB0); // End Sequence
Serial1.write((byte)0xB3);
}

```